Alma Mater Studiorum
Università degli studi di Bologna

seconda facoltà di ingegneria
Corso di Laurea in Ingegneria Informatica

Aspetti di sicurezza nel Cloud Computing

Tesi di Laurea in Sistemi Distribuiti

Relatore:
Chiar.mo Prof.
Andrea Omicini

Presentata da:
Francesca Collina

Correlatore:
Dott. Ing.
Nazzareno Pompei

II Sessione
Anno Accademico 2011/2012

A chi lotta per ottenere ciò che vuole...

Introduzione

Il continuo aumentare delle richieste di servizi informatici necessari per soddisfare il singolo utente, ma anche il business, ha portato, con il passare del tempo, ad un incremento del numero di server utilizzati nei data center delle organizzazioni e ad un maggiore uso della virtualizzazione come tecnica di memorizzazione.

Il risultato di questa recente evoluzione ha contribuito all'introduzione di un nuovo tipo di tecnologia: il Cloud computing.

Esso porta ad un aumento della capacità di storage ed elaborazione, ma in particolare la computazione passa da singole macchine, distribuite e indipendenti, a posizioni unificate e centralizzate.

Il Cloud computing è un genere di virtualizzazione, ossia una delocalizzazione e astrazione di risorse di elaborazione, di memorizzazione e di contenuti, distribuiti e raggiungibili attraverso la rete.

Si conoscono vari aspetti del Cloud, a livello di interoperabilità con lo user, a livello di architettura e di deployment.

Questa nuova tecnologia porta ad ottenere maggior flessibilità, ottimizzazione delle risorse e contenimento dei costi, ma soprattutto introduce il modello "pay-as-you-go", ovvero poter sfruttare un servizio solo per il minimo necessario di cui l'utente abbia bisogno.

L'applicazione di tale tecnologia ha evidenziato la presenza di alcuni problemi riguardo la sicurezza, ovvero l'accesso ai dati e ai servizi, legati soprattutto alla sua implementazione nei meccanismi di connessione alla rete e trasporto delle informazioni, ma anche riguardo la privacy, in particolare

inerenti a questioni politico legali.

L'obiettivo di questa tesi è di mostrare una panoramica sugli aspetti principali del Cloud computing, soffermandosi principalmente sugli aspetti di sicurezza e privacy.

Viene messa in evidenza la gestione dei dati, analizzando sia l'aspetto informatico stesso, ma anche l'aspetto giurisdizionale di alcune situazioni.

Nel primo capitolo, si mette in luce la nascita del Cloud computing, studiando la sua intera evoluzione. Dal *Grid Computing*, all'*Utility Computing* fino al Cloud come Software as a Service.

Nel secondo capitolo, viene presentato il Cloud computing in tutti i suoi aspetti. Le caratteristiche principali, l'architettura e i modelli di deployment.

Nel terzo capitolo, si entra nel merito del tema della tesi. Vengono descritti i principali problemi di sicurezza che si possono presentare nell'uso del Cloud. Concernenti a questi, vengono mostrati alcuni aspetti giuridici nel rapporto tra client e provider.

Nel quarto capitolo, si va nello specifico del "contratto" tra client e provider. Aspetti positivi e negativi di questa gestione.

Nel quinto capitolo, si ha un esempio concreto di quello che può essere un accordo tra le parti.

Si parla di *Service Level Agreement*, del quale si cerca di spiegare le principali caratteristiche, la struttura e le sue funzioni.

Infine, viene studiato come, in alcuni casi, è gestita la sicurezza in *Ajaxplorer*, software open source.

Indice

Capitolo 1

Nascita e sviluppo del Cloud computing

Il Cloud computing nasce come evoluzione di tecnologie preesistenti. Si inizia parlando di grid computing o sistemi grid, che rappresentano un'infrastruttura di calcolo distribuito, utilizzati per l'elaborazione di consistenti quantitativi di dati. Essi permettono la condivisione coordinata di risorse all'interno di un'organizzazione virtuale.

L'*utility computing*, invece, è l'inizio di quello che oggigiorno viene definito servizio "on-demand" ,ovvero è un raggruppamento di risorse di calcolo, il cui vantaggio risiede nei bassi costi iniziali di acquisizione di risorse del computer.

In seguito, si arriva a parlare di SaaS, ovvero *Software as a Service*, visto come un software che appartiene, che è distribuito e amministrato da remoto, da uno o più provider. Esso consente una condivisione di processi applicativi e risorse di storage in ambienti uno-a-molti, con delle basi di "pay-per-use", o sottoscrivendo un contratto.

Infine il *Cloud computing*, il quale si crede sia oggi la tecnologia che possa rimodellare l'industria IT.

1.1 Grid Computing

"A computational GRID is a hardware and software infrastructure that provides dependable, consistent, pervasive, and inexpensive access to high-end computational capabilities."

La Grid è un ambiente persistente, che rende possibile realizzare applicazioni che integrino risorse di strumentazione, di visualizzazione, di calcolo e di informazione provenienti da dominii amministrativi diversi e geograficamente distribuiti.

Principalmente essa coordina e condivide risorse computazionali di diverso tipo che non sono sottoposte a un controllo centralizzato, usa protocolli, librerie e interfacce standard, aperte e "general-purpose" e garantisce Quality of Service (QoS).

In particolare dà la possibilità di condividere in modo coordinato risorse all'interno di un'organizzazione virtuale dinamica e multi-istituzionale (Virtual Organization, VO).

1.1.1 Principali Tipi

Esistono tre principali tipi di Grid:

- *Grid computazionale*
- *Grid dati*
- *Grid applicazioni e/o servizi*

La *Grid computazionale* è vista come l'aggregazione di risorse di calcolo provenienti da dominii di sicurezza e gestione differenti,è finalizzata a fornire, ad un insieme di utenti, potenza di calcolo "on-demand", in modo disaccoppiato dalla provenienza.

La *Grid dati* può essere considerate una forma evolutiva del web: nasce per contenere una grande quantità di dati distribuiti in dominii differenti.

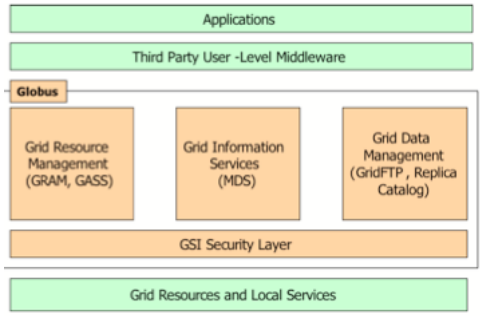

Figura 1.1: Struttura Grid Computing

Infine la *Grid applicazioni e/o servizi*, è uno degli aspetti più innovativi, in quanto assume il ruolo di *Application Service Provider (ASP)*, cioè prende in locazione un certo tempo di esecuzione di una applicazione su di un server remoto e consente anche di realizzare al suo interno nuove specifiche applicazioni.

1.1.2 Problematiche principali

Il Grid computing presenta alcuni considerevoli problemi.

Innanzitutto, essendo una tecnologia che sfrutta le potenzialità di internet, essi possono sorgere proprio dalla banda e latenza non note a priori. Ciò dipende dal traffico di dati che contiene, ma soprattutto dal cambiamento nel tempo della connettività.

Un altro fattore che influisce negativamente può essere la necessità di partizione e aggregazione delle varie macchine virtuali in modo dinamico.

A livello di gestione poi, si notano aspetti riguardanti la molteplicità di politiche che non necessariamente sono compatibili fra loro, la non conoscenza dell'affidabilità dei nodi a priori e, per finire, aspetti tipici sulla sicurezza.

1.2 Utility Computing

È un servizio che permette la distribuzione, l'amministrazione e la scalabilità on line dei servizi offerti all'utente, il quale paga per le risorse che consuma. Gli utenti vogliono essere a conoscenza di cosa ospita ogni server e controllare le risorse direttamente.

1.3 Grid computing vs Utility computing

Sia il Grid computing sia l'utility computing sono basati su reti eterogenee, hanno una struttura distribuita geograficamente e anche le stesse risorse sono eterogenee.

La principale differenza tra le due tecnologie consiste nella gestione dell'*inter*-organizzazione della Grid e dell'*intra*-organizzazione dell'utility. Quando si parla di inter-organizzazione si nota che le politiche di gestione sono diverse, così come i requisiti di sicurezza, si hanno problemi di SLA (Service Level Agreement) e di accounting.

Tutto ciò non esiste nell' inter-organizzazione dell'utility computing, dove avviene l'esatto contrario.

1.4 Software as a Service

Un sempre maggior numero di applicazioni software di utilizzo corrente sono applicazioni Web, quindi di fatto applicazioni SaaS, eseguite su computer di terzi, nonchè mantenute da questi ultimi, disponibili praticamente "on the Cloud".

Il termine SaaS viene spesso usato indistintamente al posto di Cloud computing, ma questi due termini non vanno confusi: è facilmente desumibile come il Cloud computing sia un concetto ben più esteso, che riguarda un nuovo paradigma computazionale e che comprende diverse nuove categorie di servizi, tra le quali anche SaaS. Si può quindi affermare che un'architettura

di Cloud computing è in relazione con SaaS, in quanto può essere considerata come il miglior meccanismo per erogare Software as a Service.

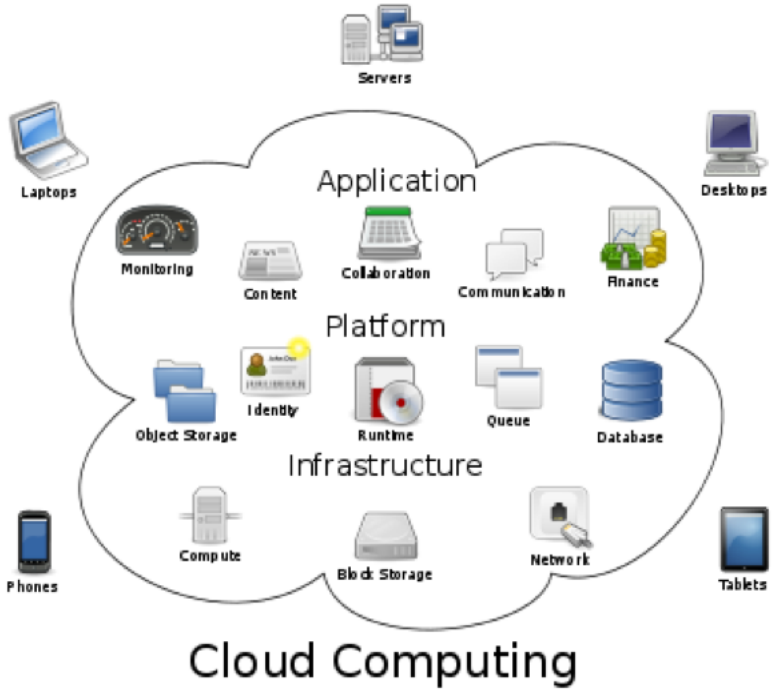

Figura 1.2: Cloud Computing

Capitolo 2

Cloud Computing

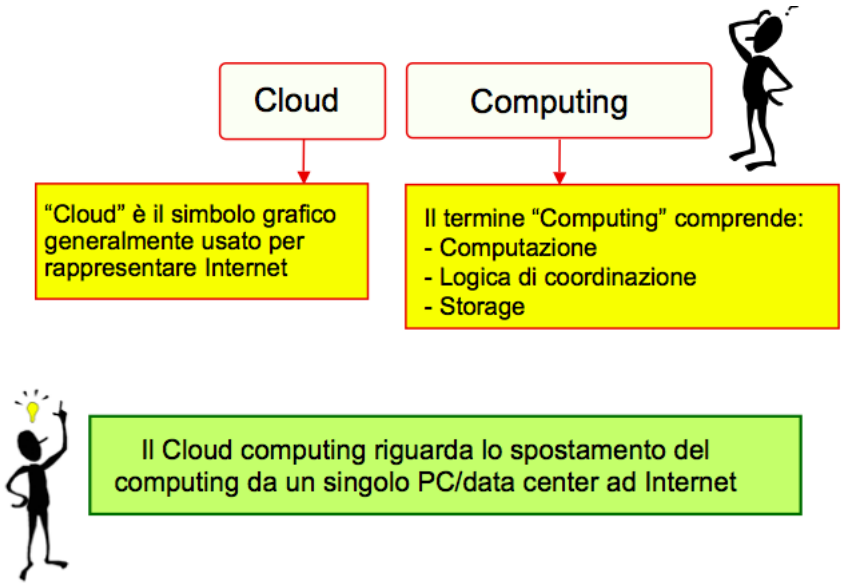

Figura 2.1: Definizione di Cloud Computing

Il Cloud computing è un paradigma in evoluzione, recentemente è emerso più volte come parola nell'ambiente di sistemi distribuiti.

2.1 Definizione

"Cloud Computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (es., networks, servers, storage, applications, and service) that can be rapidly provisioned and released with minimal management effort or service provider interaction." [1]

La definizione NIST caratterizza importanti aspetti del Cloud Computing ed è destinata a servire come mezzo per il confronto tra i servizi e le varie strategie di distribuzione, cercando anche di fornire una distinzione da ciò che è il Cloud Computing e da come utilizzarlo.

I modelli di servizi e di distribuzione definiti creano una tassonomia semplice che non porta, però, a prescrivere o a limitare qualsiasi metodo di implementazione, funzionamento ed erogazione del servizio.

2.1.1 Principali attori

Cloud Consumer: è la persona o l'organizzazione che usa servizi del Cloud provider con cui mantiene relazioni di business.

Cloud Provider: è la persona, l'organizzazione o l'ente che rende disponibili i servizi al Cloud consumer.

Cloud Carrier: è l'intermediario che fornisce connettività tra il Cloud consumer ed il Cloud provider.

[1] [U.S. NIST (National Institute of Standards and Technology)]

2.2 Caratteristiche principali

Tra le caratteristiche di base del Cloud troviamo:

1. *On-demand self-service*: un consumer, con una necessità istantanea, può usufruire di "computing recources" in modo automatico, senza dover contattare il fornitore;

2. *Broad network access*: queste "computing resources" sono utilizzate tramite la rete (es. Internet) e sono usate da vari "client application" con piattaforme eterogenee (mobile phones, laptops ecc.).

   Quello che deve essere garantito è un accesso alla rete ampio e affidabile. Oggigiorno, il continuo e maggior aumento di collegamenti ad alta banda permette l'utilizzo di servizi e di software potenzialmente localizzati in tutto il mondo;

3. *Resource pooling*: le risorse di un provider di servizi Cloud sono raggruppate in modo da riuscire a soddisfare una moltitudine di consumers, usufruendo del multi-tenancy e della virtualizzazione. Il risultato di questo "aggregare" è che l'hardware diventa invisibile al consumer, che in genere non ha il controllo o le conoscenze per gestirlo. Per esempio, il consumer non sa dove i suoi dati sono salvati nel Cloud;

4. *Rapid elasticity*: per i consumatori, le risorse di calcolo diventano immediate: non ci sono impegni e contratti diretti, in quanto questa elasticità permette sempre di avere un utilizzo pari alle necessità. Inoltre la risorsa fornita appare al consumer infinita, perciò il consumo può aumentare rapidamente per soddisfare le richieste di picco in qualsiasi momento;

5. *Measured service*: nonostante le "computing resources" siano raggruppate e condivise da molti consumers, l'infrastruttura Cloud è in grado di sfruttare appropriati meccanismi, per misurare l'uso di queste resources dei singoli consumer e poter garantire il costo in base all'uso che si fa del servizio.

2.3 Architettura del Cloud computing

La fase di passaggio verso il Cloud computing sta delineando nuove categorie di servizi IT, che consentono di creare applicazioni, database e servizi Web di qualunque genere, garantendo storage, backup, data replication, data protection, security, ecc.

Tra le principali categorie ricordiamo *Hardware as a Service (HaaS), Software as a Service (SaaS), Platform as a Service (PaaS), Infrastructure as a Service (IaaS) e Data storage as a Service (DaaS).*

Grazie all'impiego di esse è possibile evitare ingenti costi infrastrutturali per l'acquisto, la manutenzione, il supporto e/o l'aggiornamento delle stesse infrastrutture informatiche.

Come descritto, il Cloud computing può essere esteso attraverso i seguenti cinque "service models":

1. Hardware as a Service (HaaS);
2. Software as a Service (SaaS);
3. Platform as a Service (PaaS);
4. Infrastructure as a Service (IaaS);
5. Data storage as a Service (DaaS).

2.3.1 Hardware as a Service (HaaS)

HaaS risulta lo strato più basso nella struttura del Cloud. Quest'ultimo si basa principalmente sulla creazione di *Virtual Machine (VM)* nei vari server, dove si hanno i data center di elevate dimensioni. L'utente finale non nota i problemi inerenti a questo strato, può solo essere a conoscenza della VM che vi è, senza però sapere dove è localizzata a livello di server.

2.3.2 Software as a Service (SaaS)

"The capability provided to the consumer is to use the provider's applications running on a Cloud infrastructure." [2]

Il termine si riferisce alla fornitura di un applicativo in modalità centralizzata e accessibile via Web. La caratteristica comune di questa tipologia di servizio è quella di fornire un applicativo che sia condiviso tra tutti i clienti a meno di funzionalità opzionali.

I consumer realizzano le proprie applicazioni in un ambiente "hosting", al quale si accede, attraverso il network, da vari client (es. web browser, PDA, ecc.) sfruttando applicazioni caratteristiche. Questo strato fornisce il software come servizio accessibile, senza necessità di installazioni sulla singola macchina. Nella maggioranza dei casi, una singola istanza dell'applicativo gestisce clienti diversi, pur garantendo la separazione logica dei dati di ciascun cliente.

Il SaaS spesso impiega un sistema architetturale "multi-tenancy", ovvero, le diverse applicazioni dei consumers sono organizzate in un singolo ambiente, per poter ottimizzare la velocità, la sicurezza e il mantenimento.

Questo strato semplifica il lavoro sia a livello del provider, il quale avrà meno problemi di gestione e sia per il consumer, il quale può permettersi di accedere allo stesso ambiente di lavoro da qualsiasi dispositivo e da qualsiasi luogo.

[2][U.S. NIST (National Institute of Standards and Technology)]

Infine i servizi applicativi offrono un'interfaccia via Web Services, che permette l'integrazione e l'interoperabilità con altri applicativi e che quindi consente di sviluppare nuove applicazioni, seguendo i principi *SOA (Service Oriented Architecture)*.

Alcuni esempi di SaaS : SalesForce.com, Google Mail, Google Docs.

2.3.3 Platform as a service (PaaS)

"The capability provided to the consumer is to deploy onto the Cloud infrastructure consumer-created or acquired applications created using programming languages and tools supported by the provider." [3]

PaaS è una piattaforma di sviluppo che supporta il "Software Lifecycle", rendendo disponibile, via Web, tutti quegli strumenti e prodotti utilizzati nello sviluppo e delivery di nuovi servizi applicativi.

Ad esempio, sono piattaforme che rendono disponibili strumenti di sviluppo come workflow, di creazione di interfacce web, database integration, storage, integrazione di web-service.

La principale differenza tra PaaS e SaaS è che SaaS offre esclusivamente la sola applicazione, mentre il PaaS supporta una piattaforma di sviluppo in cui si ha sia l'applicazione completata sia quella in-progress.

Il PaaS richiede un'infrastruttura e include anche un ambiente di sviluppo (es. tools).

Alcuni esempi di PaaS: Google AppEngine, Microsoft Azure e Force.com

[3][U.S. NIST (National Institute of Standards and Technology)]

2.3.4 Infrastructure as a Service (IaaS)

"The capability provided to the consumer is to provision processing, storage, networks, and other fundamental computing resources where the consumer is able to deploy and run arbitrary software, which can include operating systems and applications." [4]

IaaS viene definito come lo strato che fornisce risorse fondamentali per gli strati superiori. Si riferisce, in particolare, alla modalità di offrire come servizi infrastrutturali, risorse di elaborazione, memoria e comunicazione, come macchine virtuali, CPU, memoria, schede LAN, apparati di rete e loro configurazioni, servizi di backup.

Il Cloud consumer usufruisce direttamente dell'infrastruttura IT (processing, storage, networks, e altri fondamentali risorse). La virtualizzazione è ampiamente sfruttata nel IaaS al fine di integrare/scomporre le risorse fisiche in una metodologia ad-hoc per soddisfare la crescente o la riduzione della domanda di risorse da parte dei consumer. La strategia fondamentale della virtualizzazione è quella di creare Virtual Machines indipendenti, chiamate anche istanze, le quali sono isolate dall'hardware sottostante e da altre Virtual Machine. Questa strategia mira a trasformare l'architettura software di applicazione in modo che più instanze (da diversi consumers) riescano a girare in una singola applicazione.

Alcuni esempi di IaaS: Amazon's EC2, Eucalyptus, OpenNebula.

2.3.5 Data storage as a Service (DaaS)

La distribuzione di virtualizzazione on demand diventa un Cloud service separato. DaaS può essere visto come un tipo speciale di IaaS. La motivazione è che sistemi di database sono spesso legati a un costo di un server o di una licenza software. DaaS permette al consumer di pagare quello di cui concretamente usufruisce, più che pagare l'intera licenza di un intero database.

[4][U.S. NIST (National Institute of Standards and Technology)]

Alcuni esempi di DaaS: Amazon S3, Google BigTable, Apache HBase.

2.3.6 Business Process Layer (BPaaS)

A volte conosciuto con altri nomi, ad esempio *Enterprise-as-a-Service*, questo livello si inserisce all'interno della tematica più ampia del business process management, che è un approccio strutturato basato su metodi, politiche, metriche per gestire ed ottimizzare continuamente le attività e i processi di un'organizzazione. Questo avviene per mezzo di un layer intermedio, in grado di consentire una suddivisione netta tra le regole e la logica applicativa, vale a dire un motore di esecuzione; per questo motivo può essere visto come un'estensione del livello precedente, in quanto è in pratica un software spinto fino a livelli massimi di configurabilità. Tutto ciò, in una visione Cloud, può essere spostato al di fuori del controllo dell'utente, che potrà accedere ai servizi attraverso un'interfaccia web e tramite architetture Web oriented.

2.3.7 Principali aspetti in comune

Sicuramente queste tipologie di servizi, prestano e offrono caratteristiche comuni che possono essere analizzate:

- la facilità dell'accesso al servizio, tramite semplici interfacce web, porta a poter gestire un'auto-configurazione del servizio richiesto (self-provisioning);
- la disponibilità di un servizio fruibile da clienti diversi (multitenancy), poichè i clienti condividono le stesse risorse hardware o software, pur mantenendo una separazione logica o fisica dei dati;
- minor spese di risorse, in quanto rese disponibili da chi fornisce il servizio;
- accesso ai servizi da remoto mediante interfacce che sono indipendenti dal tipo di device utilizzato;

- aggiornamenti software o nuove funzionalità sono resi disponibili a tutti i clienti del servizio, in maniera trasparente e normalmente senza ulteriori aggravi di spesa da parte degli utenti;

- possono essere resi disponibili dei servizi che sono il risultato di un'integrazione di differenti servizi disponibili nei vari Cloud, servizi che possono essere configurati anche dinamicamente in base ad esempio a criteri di scalabilità, di disponibilità delle risorse, o ai costi di erogazione dei servizi stessi.

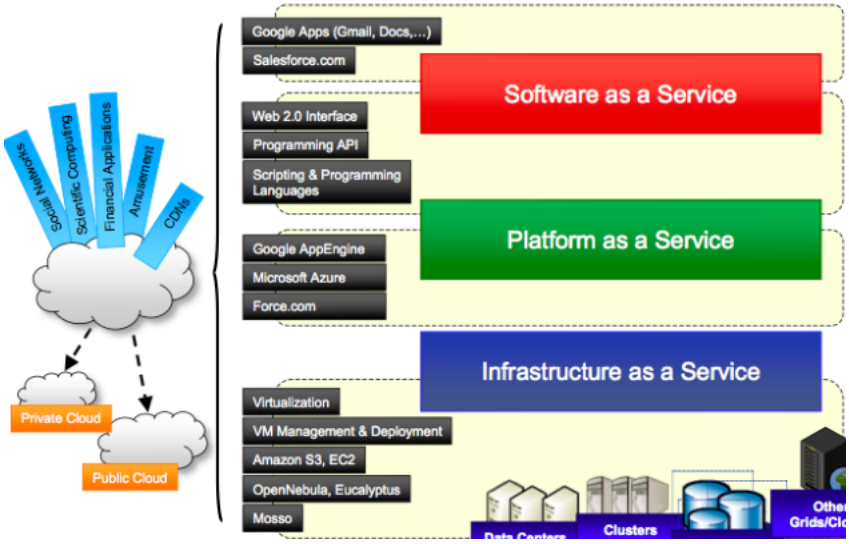

Figura 2.2: Services Models

2.4 Modelli di deployment di Cloud

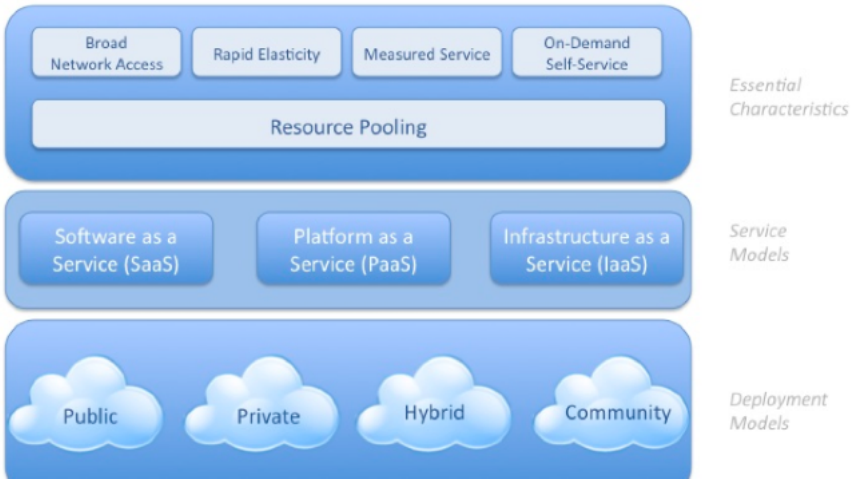

Figura 2.3: NIST Visual Model of Cloud Computing Definition

La descrizione precedente prevedeva una prospettiva del Cloud dal punto di vista dell'utilizzatore, ora, invece, vediamo le distinzioni dal punto di vista del provider, partendo da quattro modelli:

1. Private Cloud;
2. Public Cloud;
3. Community Cloud;
4. Hybrid Cloud.

2.4.1 Private Cloud

L'infrastruttura Cloud è utilizzata esclusivamente all'interno di una singola organizzazione e gestita dall'organizzazione stessa o da un terzo.

La motivazione per impostare un Cloud privato all'interno di una organizzazione ha diversi aspetti: innanzitutto, per massimizzare e ottimizzare l' utilizzo delle attuali risorse interne, in secondo luogo, per problemi di sicurezza, tra cui la riservatezza dei dati. In terzo luogo, poi, risulta di notevole importanza a livello di costo, basso, al contrario di quello del trasferimento dati dalla locale infrastruttura IT a un Cloud pubblico, che appare piuttosto elevato.

2.4.2 Public Cloud

Questo tipo di Cloud è la forma più dominante di distribuzione. Nel public Cloud il provider di servizi ha la piena proprietà e padronanza dell'intera struttura, con la propria politica, il proprio valore e profitto, e il proprio costo. La connessione che si sfrutta, il piú delle volte, può anche essere una VPN, per collegarsi direttamente con la rete aziendale.

Esempi di public Cloud popolari sono: Amazon EC2, S3, Google AppEngine, e Force.com

2.4.3 Community Cloud

Si parla di community Cloud quando diverse organizzazioni insieme costituiscono e condividono la medesima infrastruttura, così anche le politiche, i requisiti e i valori.

2.4.4 Hybrid Cloud

L'Hybrid Cloud è una combinazione di due o più Cloud (privata, comunity o pubblica) rimanendo entità uniche, ma sono tenute insieme da tecnologie standardizzate che permettono la gestione dei dati e la portabilità delle applicazioni. Le organizzazioni utilizzano il modello ibrido per ottimizzare le proprie risorse, per aumentare le loro competenze di base usufruendo gradualmente del Cloud. I servizi (IaaS,PaaS ecc) sono erogati da un'infrastruttura distribuita tra i data center del consumer e del provider. Il problema principale di questo tipo di Cloud è capire come distribuire nel modo migliore i carichi elaborativi, fra private e public e come integrare le due architetture per mezzo di tecnologie stardard o proprietarie.

È interessante notare come *Amazon Web Service (AWS)* ha recentemente lanciato un nuovo tipo di modello di distribuzione *Virtual Private Cloud (VPC)*, un "ponte" sicuro e senza soluzione di continuità tra un'organizzazione di infrastruttura IT esistente e il Cloud Amazon pubblico. Questa si posiziona come un misto fra Private Cloud e Public Cloud.

Viene considerato pubblico perchè usa le risorse di calcolo comuni da parte di Amazon. Tuttavia è anche privata per due ragioni: in primo luogo, il collegamento IT legacy e il Cloud è garantito da una rete privata virtuale, sfruttando, in questo modo, il vantaggio di sicurezza del Cloud privato. Non a caso, tutte le politiche di sicurezza aziendale si applicano ancora alle risorse sul Cloud anche se si trova su un Cloud pubblico. In secondo luogo, AWS dedica una serie di risorse "isolate" per la VPC. Questo non significa che gli utenti siano costretti a pagare queste risorse "isolate"; essi, infatti, possono continuare a usufruire del "pay-per-use".

VPC rappresenta un perfetto equilibrio tra controllo (Private Cloud) e flessibilità (Public Cloud).

2.5 Vantaggi e svantaggi

2.5.1 Vantaggi

Come abbiamo visto il Cloud computing è in grado offrire diverse caratteristiche di servizio che risultano utili per l'user finale.

- Infrastruttura "agile": i servizi IT, basati su Cloud, sono sempre più in grado di sostenere una forza lavoro sempre maggiore, permettendo ai consumatori di poter accedere alle proprie risorse ovunque essi si trovino.

- Risposte rapide: il Cloud permette di potersi espandere facilmente, potendo soddisfare le esigenze.

  Non è più necessario dover acquistare, da parte di aziende, di risorse di calcolo, in quanto il Cloud garantisce variazioni e richieste "on-demand" della potenza di elaborazione e di calcolo.

- Minor costi: non si hanno più costi eccessivi di manutenzione delle infrastrutture IT.

  Un servizio Cloud è disponibile "on-demand", seguendo la logica del "pay-per-use", ovvero acquistando solo quello di cui si usufruisce. Si riducono sprechi di risorse e, di conseguenza, la spesa.

  Se si necessita di una maggior potenza di calcolo, è possibile adottare nuovi applicativi senza dover affrontare investimenti onerosi.

- Minor guasti al sistema: con l'adozione del Cloud computing, si evitano problemi di interruzione del servizio. Ciò significa, che se si verifica un guasto ad un singolo nodo, si continua ad avere comunque funzionalità dal servizio. I tempi di ordinazione, realizzazione, installazione e configurazione sono estremamente ridotti. Così come i costi di manutenzione e riparazione.

- User friendly: la sua interfaccia risulta alquanto semplice, che nella maggioranza dei casi è Web, porta ad un facile rapporto tra l'user e l'applicazione Cloud in sè.

2.5.2 Svantaggi

Sicuramente, uno dei problemi principali a cui si deve far fronte, riguarda la privacy e la sicurezza dei propri dati. Questa perplessità afferisce alla "riservatezza ed alla legalità", non solo dei dati personali, ma ancora di più dei dati sensibili. L'uso di servizi Cloud comporta l'accettazione di termini d'uso e proprio i dati vengono trattati secondo i termini acettati.

Altri problemi riguardanti il Cloud, nella maggior parte dei casi, possono derivare dalla gestione della "rete". La velocità, il rischio di perdere pacchetti dati nel trasferimento, ecc.

Può capitare che si arrivi, anche, ad una saturazione di accesso al servizio, dovuta alle tante richieste, con conseguenti problemi di accesso.

2.6 Cloud computing vs Grid computing

Esistono evidenti somiglianze tra il Cloud computing e il Grid computing descritte nel capitolo precedente. Grid e Cloud computing si sono imposti soluzioni di rilievo nell'ambito di gestione dei dati e delle risorse di calcolo distribuito, portando ad un livello più avanzato la gestione delle collaborazioni e la qualità del flusso informativo.

Tuttavia tra Grid e Cloud si rilevano differenze significative:

- la grid sottolinea la "condivisione delle risorse" per creare un'organizzazione virtuale; il Cloud è spesso di proprietà di una singola organizzazione fisica, la quale alloca le risorse per le diverse istanze in esecuzione;

- la grid mira a fornire la massima potenzialità e capacità di calcolo per compiere compiti abbastanza consistenti, attraverso la condivisione delle risorse; il Cloud punta a soddisfare soprattutto in tempo reale le richieste dei diversi utenti;

- la grid cerca di ottenere la massima elaborazione; il Cloud, allo stesso tempo, cerca di ottimizzare la capacità di elaborazione.

Capitolo 3

Sicurezza e rischi nel Cloud Computing

" Security controls in cloud computing are, for the most part, no different than security controls in any IT environment. However, because of the cloud service models employed, the operational models, and the technologyies used to enable cloud services, cloud computing may present different risks to an organization than traditional IT solutions." [1]

Come visto nei capitoli precedenti, il Cloud computing ha dato al mondo IT un grande contributo a livello di possibilità e di servizi disponibili. Si è assistito ad un incremento, in numerosità e complessità, sia delle tipologie di servizi fruibili, sia delle tipologie di fornitori di servizi, che hanno dato vita ad un grande sistema di offerte particolarmente articolato e complesso.

A fronte di questi indubbi vantaggi, ogni nuova tecnologia presenta problematiche di vario genere.

Per questo, si nota una certa cautela nell'affidarsi completamente al Cloud computing, specialmente per le aziende, le quali sentono il rischio di trasferire certi processi aziendali nel Cloud, laddove ci siano informazioni di natura confidenziale o non tutelate da normative nazionali e/o internazionali.

[1] [Cloud Security Guidance di CSA (Cloud Security Alliance)]

Questa nuova tecnologia presenta diversi aspetti problematici, non solo si parla di problemi "classici" (e.g. data privacy, data lock-in, scalable storage, ecc), ma vi sono anche aspetti di tipo legale, derivanti da questioni riguardanti i fornitori di servizio, i quali hanno la necessità di disporre di infrastrutture in vari luoghi geografici.

In questo modo, il cliente è esposto a diverse giurisdizioni, in base al luogo geografico, quindi sottoposto a normative specifiche.

3.1 Access control

I clienti interessati hanno il diritto e la possibilità di sapere dove i loro dati si trovano ed eventualmente poter decidere se tenerli o cancellarli.

La domanda spontanea che sorge è: ma si è sicuri che una volta cancellati, i dati siano stati veramente eliminati? Chi assicura tutto ciò?

Nel momento in cui si pretende uno spostamento di dati, una cancellazione o modifica, ci si deve affidare al provider, sapendo che ha il diretto accesso alle risorse. Molti problemi sono causati, per esempio, dalla diminuzione del controllo da parte del titolare sui propri dati e sul trattamento di questi a carico del provider, in quanto i dati non risiedono sotto il diritto del controllo del titolare.

Inoltre, si ha anche un aumento di esposizione dei dati critici, dovuta alle caratteristiche architetturali condivise, tipiche del Cloud computing.

Un altro aspetto rilevante è l'aumento dell'utilizzo di reti pubbliche per l'accesso a risorse remote.

Per evitare queste problematiche, causando violazioni di privacy o semplicemente perdita di dati, si cerca, per esempio, di limitare l'accesso ai dati, solo in caso di necessità. Controllare in background coloro che lavorano in questo ambito, formare il personale addetto sulla base di certi requisiti, fornire un'organizzazione in materia di accesso ai set di dati, da parte del personale.

Questo tipo di problema si basa solo esclusivamente sulla "lealtà" di un singolo individuo, non ha soluzioni o cause prettamente tecnologiche.

Al giorno d'oggi, però, è sempre più frequente questa minaccia e sempre più difficile è contenere questi rischi, trovando metodi e risoluzioni per far scomparire il pericolo.

È opportuno capire i principali rischi, seguendo il punto di vista classico di "localizzazione del dato" in ottica privacy:

- *data-at-rest*: in questa sezione si pone attenzione alla cancellazione dei dati. Non sempre si ha la garanzia dal provider che i dati siano effettivamente eliminati in maniera sicura, soprattutto nel caso in cui si abbia una rescissione del contratto o una riallocazione di risorse IT.

- *data-in-transit*: questo termine specifica che i dati vengono memorizzati entro il perimetro del provider, a prescindere dal tipo di dati di cui si tratta.

  In questo caso specifico, si assiste al passaggio di dati dal cliente al fornitore, dando origine ad un flusso informativo tra due ambienti.

  Questo transito di informazioni deve essere regolato con grande attenzione, a livello tecnologico, nelle responsabilità e quindi anche a livello contrattuale. Spesso i dati in transito vengono sottoposti a cifratura, la quale è considerata la miglior "arma" per ridurre i rischi in questa delicata fase di passaggio di informazioni;

- *data-in-process*: in questo caso, come la parola stessa suggerisce, ci si riferisce al trattamento dei dati, i quali non risultano però cifrati, a differenza del caso precedente, e quindi sono vulnerabili per ogni tipo di pericolo, anche non Cloud;

3.1.1 Identity

Identità consiste in un insieme di informazioni associate a una specifica entità. Le piattaforme cloud dovrebbero garantire e fornire un robusto sistema di gestione di queste.

Infatti esso dovrebbe includere, per esempio: *identity provisioning e deprovisioning, identity information privacy, identity federation, authentication and authorization.*

Inoltre, tale sistema dovrebbe adottare standard esistenti come:

SPML [2], SAML [3], OAuth [4] e XACML [5], per garantire una sicura identità fra entità interagenti all'interno di dominii diversi e piattaforme Cloud.

[2]Il Service Provisioning Markup Language: è lo standard aperto per l'integrazione e l'interoperabilità delle richieste di servizio di provisioning. Provisioning è l'automazione di tutte le fasi necessarie per gestire (installare, modificare e revocare) i diritti di accesso dell'utente o di sistema o dati relativi ai servizi di pubblicazione elettronica.

[3]Security Assertion Markup Language: è lo standard per lo scambio di dati di autenticazione e autorizzazione tra domini di sicurezza. SAML è un protocollo basato su XML che utilizza i token di protezione contenente informazioni da trasmettere tra un provider di identità e di un servizio web.

[4]OAuth è un protocollo aperto che permette l'autorizzazione di API di sicurezza con un metodo standard e semplice sia per applicazioni portatili che per pc fisso e web. In pratica permette all'utente di dare l'accesso alle sue informazioni presenti, ad esempio, su un sito detto service provider, ad un altro sito, chiamato consumer, senza però condividere la sua identità

[5]eXtensible Markup Language Access Control: è lo standard che definisce un linguaggio dichiarativo di controllo di accesso attuato in XML, e di un modello di trasformazione che descrive come valutare le richieste di autorizzazione in base alle regole definite nei criteri.

3.1.2 Data lock-in

"Cloud lock-in is a situation in wich a cloud user or customer will experience issues when he or she decides to switch to another Cloud vendor, due to the complexity of the switching process".

Questo problema consiste nell'estrarre dati da un Cloud, senza perdere informazioni personali, potendo riutilizzarli a sua volta su un altro Cloud. La difficoltà di estrarre dati dal Cloud impedisce a organizzatori e aziende di usufruirne, in quanto aumenterebbe i costi e i problemi di affidabilità.

La principale soluzione a questa problematica sarebbe nel personalizzare le API, in modo che uno sviluppatore possa implementare servizi SaaS e gestire dati attraverso più provider di cloud computing, senza rischiare di perdere informazioni, nel caso siano correlati con altri Cloud.

In particolare, la standardizzazione di API consentirebbe un nuovo modo di utilizzo del software, in quanto potrebbe essere utilizzato in un centro dati interno e in un cloud pubblico.

3.1.3 Storage Location

Nella maggior parte dei casi, il fornitore di servizi può memorizzare i vari dati e/o file forniti dal cliente sul server a lui più conveniente, per la gestione del servizio. Ciò porta, talvolta, ad un trasferimento dei dati in server fuori dalla propria regione, o anche fuori dal proprio paese d'origine.

Questo tipo di organizzazione comporta problemi riguardanti il rispetto di regole e leggi, alle quali le stesse organizzazioni che richiedono il servizio devono sottostare.

Per evitare questo rischio, si dovrebbe cercare di ottenere maggior informazioni riguardo questi aspetti, ancora prima di prendere un accordo con il fornitore di servizi.

Ad esempio, valutare le leggi e i regolamenti che le organizzazioni devono seguire, quindi capire se è possibile autorizzare il trasferimento dei dati al di fuori della propria zona geografica evita di violare una giurisdizione specifica.

3.1.4 Problematiche nei "Services Models"

- **Problemi IaaS**

  Come spiegato nel secondo capitolo, l'IaaS si base soprattutto sulla virtualizzazione come strumento principale, assieme all'uso di Virtual Machine. Proprio in quest'ultima si annidano varie e serie minacce alla sicurezza, quali malware e virus.

  La sicurezza delle Virtual Machine è la responsabilità dei consumatori Cloud. Ogni consumatore può utilizzare i propri controlli di sicurezza in base alle proprie esigenze.

  La protezione di dati, nelle Virtual Machine, a differenza dei server fisici, crea problemi anche in stato di offline.

  In molti casi, il file immagine della Virtual Machine può essere compromesso da codici maligni, come può essere anche sottratto al consumatore.

  Inoltre, a volte, i dati originari relativi ad un consumatore, salvati in questo tipo di sistema, possono rimanere in queste VM, che a loro volta vengono riutilizzate da terzi.

  Si devono tener presente anche i rischi derivanti dalla rete virtuale, ovvero la condivisione delle infrastrutture dello stesso server o delle stesse reti fisiche che può aumentare la possibilità di sfruttare le vulnerabilità dei server DNS [6], DHCP [7], IP protocol [8].

[6] Domain Name System: è un sistema utilizzato per la risoluzione di nomi dei nodi della rete (host) in indirizzi IP e viceversa.

[7] Dynamic Host Configuration Protocol: è un protocollo di rete di livello applicativo che permette ai dispositivi o terminali di una certa rete locale di ricevere dinamicamente, ad ogni richiesta di accesso ad una rete IP, la configurazione IP necessaria per stabilire una connessione.

[8] Internet Protocol (IP): è il protocollo di rete su cui si basa la rete Internet.

- **Virtualizzazione**

  Le Virtual Machine hanno confini virtuali rispetto ai server fisici, ovvero non vi è isolamento fisico tra le risorse della VM, infatti esse coesistono sullo stesso server fisico, usufruiscono della stessa CPU, memoria, I/O, scheda di rete e altri componenti.

  La responsabilità dei confini VM è data al Cloud provider, mentre quella inerente all'*Hypervisor*, ovvero il principale controller di ogni accesso alla VM, è nelle mani del fornitore di servizi Cloud.

  Hypervisor o Virtual Machine Monitor è il software che realizza la condivisione stessa. Esso opera in modo trasparente senza pesare alla propria attività sul funzionamento dei sistemi operativi.

  L'adozione di questo tipo di tecnologia porta sì a molti progressi, ma allo stesso tempo la virtualizzazione crea un ambiente di sicurezza, senza dubbio, più complesso e rischioso.

  Alcune vulnerabilità e questioni di sicurezza:

  - *Remote management vulnerabilities*: alcuni Hypervisor utilizzano, a loro volta, sistemi per gestire le VM. Questo comporta nuove vulnerabilità, tra cui *cross-site scripting* [9], *SQL injection* [10] ecc.
  - *Virtual machine based Rootkit*: il concetto di *rootkit* è apparso nel mondo UNIX. Un rootkit è un insieme di strumenti (programmi) che consentono l'accesso a livello di amministratore a un computer o a una rete di computer. Se un rootkit compromette l'hypervisor, è possibile ottenere il controllo dell'intera macchina fisica.

    Rootkit tipici: Blue Pill e SubVirt.

[9] Cross-site scripting: permette ad un hacker di inserire od eseguire codice lato server al fine di attuare un insieme variegato di attacchi quali ad esempio: raccolta, manipolazione e reindirizzamento di informazioni riservate, visualizzazione e modifica di dati presenti sui server, alterazione del comportamento dinamico delle pagine web ecc.

[10] SQL injection: questa tecnica sfrutta l'inefficienza dei controlli sui dati ricevuti in input ed inserisce codice maligno all'interno di una query SQL.

– *Revert to snapshots problem*: è un meccanismo per consentire all'amministratore di fare uno "snapshot" della macchina in un certo punto e ripristinare in caso di necessità.

  Lo "snapshot" porta anche alcuni problemi di sicurezza, riguardanti certe politiche, come riattivazioni degli account precedentemente disabilitati e aggiornamento password.

- **Virtual Network**

  L'interconnettività è uno dei maggiori problemi di sicurezza del Cloud computing.

  Uno dei modi migliori per evitare questo tipo di problema è isolare ogni VM, utilizzando un collegamento ad un canale fisico dedicato per ogni host-VM.

  Tuttavia, le modalità di configurazione della rete virtuale per collegare le VM, comuni per la maggior parte degli hypervisor, sono basate nell'uso di *bridge e route*.

  In queste modalità, le prestazioni di comunicazione tra le VM risultano non compatibili quando le VM sono in esecuzione sullo stesso host.

  Come risultato, l'isolamento rischia di essere attaccato facilmente.

  Alcune vulnerabilità nella rete virtuale:

  – *Sniffing virtual network*: il *bridge* [11] gioca un ruolo di hub [12] "virtuale".

    Tutte le VM condividono l'hub virtuale per comunicare in rete, dove una VM è in grado di intercettare pacchetti dati durante il trasferimento.

  – *Spoofing virtual network*: il *route* svolge un ruolo di *switch* [13] "virtuale". Quest'ultimo utilizza un'interfaccia virtuale dedicata per collegare ciascuna VM.

[11] È un dispositivo di rete che si colloca al livello *datalink* del modello ISO/OSI e che traduce da un mezzo fisico ad un altro all'interno di una stessa rete locale.

[12] Rappresenta un concentratore, un dispositivo di rete che funge da nodo di smistamento di una rete di comunicazione dati.

[13] È un dispositivo di rete o nodo interno di rete che si occupa di commutazione a livello 2, cioè livello datalink del modello ISO/OSI di indirizzamento e instradamento all'interno di reti locali attraverso indirizzi MAC, inoltrando selettivamente i frame ricevuti verso una porta di uscita.

In questo modo, una VM può utilizzare un *Address Resolution Protocol (ARP)* [14], reindirizzando i pacchetti e quindi essere in grado di intercettare i pacchetti che vanno a certe VM, provenienti da altre macchine virtuali.

- **Problemi PaaS**

  Il PaaS si basa essenzialmente, sulla SOA [15], ovvero Service Oriented Architecture. Questo porta a ereditare tutti i problemi inerenti all'ambito SOA, come gli attacchi DOS [16] , Main in the middle attack[17], attacchi XML, ecc. Questi problemi di sicurezza sono di responsabilità divisa tra i Cloud provider, i consumatori e i fornitori di servizi Cloud.

  Il PaaS fornisce delle API che dovrebbero essere in grado di garantire sicurezza, per esempio la OAuth, per gestire l'autenticazione e l'autorizzazione.

[14] È un protocollo di rete appartenente alla suite del protocollo internet (IP) versione 4, il cui compito è fornire la "mappatura" tra l'indirizzo IP a 32bit (4byte) e l'indirizzo MAC (MAC address) corrispondente di un terminale in una rete locale ethernet.

[15] Indica generalmente un'architettura software adatta a supportare l'uso di servizi Web per garantire l'interoperabilità tra diversi sistemi così da consentire l'utilizzo delle singole applicazioni come componenti del processo di business e soddisfare le richieste degli utenti in modo integrato e trasparente.

[16] DOS: Denial of Service. Si tratta di un attacco informatico che tenta di portare il funzionamento di un sistema informatico che fornisce un servizio al limite delle prestazioni, fino a renderlo non più in grado di erogare servizi.

[17] MITM: è un tipo di attacco nel quale l'attaccante è in grado di leggere, inserire o modificare a piacere, messaggi tra due parti senza che nessuna delle due sia in grado di sapere se il collegamento che li unisce reciprocamente sia stato effettivamente compromesso da una terza parte

- **Problemi SaaS**

  Nel modello SaaS, la responsabilità dell'esecuzione e del mantenimento della sicurezza è condivisa tra il Cloud provider e il fornitore di servizi.

  Il SaaS eredita i problemi discendenti dal PaaS e IaaS, in quanto si appoggia a questi, comprendendo sia la sicurezza sui dati sia quella di rete.

  Si deve tener presente anche di "vulnerability scanning" [18].

  Nel SaaS si riscontra anche il *multi-tenancy*, il quale porta problematiche causate dalla condivisione di risorse come l'hardware e le applicazioni stesse, rischiando la perdita di informazioni.

3.2 Modelli di deployment

In relazione ai modelli di deployment, descritti nel secondo capitolo, si ha una differenziazione della sicurezza nelle varie tipologie di Cloud.

Nel modello *Public Cloud* non si ha la possibilità di variare le clausole contrattuali standard, pertanto la verifica del rispetto delle misure minime previste dalle normative privacy deve essere eseguita ex ante alla stipula del contratto.

Nel *Community Cloud*, le misure minime di sicurezza sono stabilite tra le organizzazioni che condividono lo stesso spazio di infrastruttura Cloud. Pertanto, è desumibile che i vari attori facciano riferimento a standard comunemente accettati.

Invece, nel *Private Cloud*, il client è il principale responsabile e ha il ruolo decisivo nell'attuazione delle misure minime da parte del fornitore del servizio.

[18] Uno scanner è un programma progettato per ricercare e mappare le debolezze di un'applicazione, di un computer o di una rete. Tipicamente lo scanner inizialmente cerca indirizzi attivi, porte aperte, sistemi operativi ed ogni applicazione in esecuzione. In questo processo lo scanner può tentare di violare.

Infine, il modello *Hybrid Cloud*, essendo una combinazione di modelli cloud, richiede maggior controllo nell'attuazione di misure minime di sicurezza. È opportuno che tali misure siano condivise su tutti i servizi Cloud utilizzati.

3.2.1 Tecniche di protezione

L'utilizzo di servizi/infrastrutture cloud comporta, inevitabilmente, il mancato controllo sui propri dati e su alcuni processi di gestione. La presenza di un Governance può portare ad una gestione sia a livello organizzativo, sia ad una diminuzione di rischi di tipo legali e tecnologici. È sufficiente, a volte, definire ruoli e responsabilità, in modo che ciascun incaricato possa soddisfare i requisiti richiesti da un Governance.

Per esempio, un caso in cui si potrebbero creare problemi è quando un provider si appoggia, a sua volta, ad eventuali terze parti. In questo ambito, se la gestione delle competenze non è ben organizzata, si può incorrere in una serie di ulteriori problemi.

Un altro punto su cui si pone l'attenzione è soprattutto la gestione delle utenze e dei relativi profili di accesso.

Il provider ha in possesso la totale responsabilità sul cosidetto "cloud admin", che risulta essere una potenziale minaccia, in quanto in grado di procurare danni ai dati/servizi offerti al cliente.

Tuttavia, non è un compito facile far rientrare questa gestione all'interno delle normative tassative a cui devono sottostare gli amministratori di sistema.

Per questo motivo, risulta importante e fondamentale il contratto che si viene a stipulare tra provider e consumer. Questo porta a garantire una sicurezza nell'erogazione del servizio, tenendo in considerazione l'evoluzione delle modalità in relazioni a quella della normativa.

3.2.2 Controllo e audit

Con "audit" si intende il monitoraggio su tutte le attività con la possibilità per le parti di condurre un'attività di controllo su certi oggetti e componenti.

L'osservazione del delegato è un aspetto fondamentale del processo di delega, a cui è riconducibile la nomina a responsabile dei trattamenti del fornitore di servizi Cloud. Questo tipo di controllo può assicurare di ottenere un contratto affidabile con il fornitore di servizio.

L'organizzazione che gestisce le attività di "audit" del servizio di Cloud computing deve procedere anche alla stipula del contratto di fornitura. L'adeguamento dei processi toccati dall'utilizzo di soluzioni Cloud deve portare ad una revisione delle politiche e delle procedure aziendali.

Più difficoltoso è l'audit dell'infrastruttura tecnologica e dei processi interni del fornitore. In questo caso si ha il vero e proprio problema di controllo di quanto dichiarato e "garantito" a livello contrattuale. Gli stessi fornitori iniziano ad adottare strumenti che permettono di offrire garanzie ai singoli clienti, senza essere costretti ad offrire la possibilità di sottoporre ad audit sistemi e processi.

3.2.3 Crittografia

In un ambiente Cloud, i dati sono condivisi con diversi client in rete, quindi si devono adottare opportune strategie per renderli "nascosti". La crittografia è uno strumento che offre proprio i vantaggi di affidamento minimo per il fornitore di servizi Cloud.

Si parla di:

- *Encrypting data in transit over networks*: vi è la necessità di crittografare le credenziali multi-uso, come ad esempio numeri di carte di credito, password e le chiavi private, in transito su Internet.

  Anche se le reti dei provider possono essere più sicure di Internet aperto, sono per la loro architettura fatte di tante componenti diverse e organizzazioni diverse che fanno parte del Cloud.

Pertanto, è importante proteggere le informazioni sensibili in transito anche all'interno della rete del provider. Di solito, questo può essere implementato con la stessa facilità in modalità SaaS, PaaS, IaaS.

- *Encrypting data at rest*: si intende la crittografia dei dati su disco o in un database di produzione in tempo reale, come si può proteggere contro un fornitore di servizi Cloud dannoso o maligno.

Capitolo 4

Aspetti e problemi giuridici

Il Cloud computing crea nuove dinamiche nel rapporto tra un'organizzazione e le relative informazioni, comportando la presenza di un terzo: il Cloud provider.

Questo porta a nuove sfide per la comprensione di come le leggi si applicano a una vasta gamma di scenari di gestione delle informazioni.

Un'analisi completa di questioni giuridiche, riguardo il Cloud computing, richiede la considerazione di dimensioni funzionali, giurisdizionali, e contrattuali.

- La *dimensione funzionale* consiste nel determinare quali funzioni e servizi del Cloud computing hanno implicazioni legali per i partecipanti e le parti interessate;

- la *dimensione giurisdizionale* riguarda il modo in cui i governi amministrano certe disposizioni legislative e regolamenti che impattano sui servizi di Cloud computing, sulle parti interessate dei dati coinvolti:

- la *dimensione contrattuale* coinvolge le strutture contrattuali, i termini, le condizioni e i meccanismi di applicazione attraverso i quali le parti interessate, in ambienti Cloud, possono affrontare e gestire gli aspetti legali e di sicurezza.

Il Cloud computing, quindi, porta ad un potenziale rischio giuridico elevato, creato dalla natura distribuita della "nube", rispetto alle tradizionali infrastrutture interne o di *outsourcing*.

4.1 Problematiche tra cliente e fornitore di servizi

I rischi da considerare nell'adottare il Cloud computing non riguardano solo la sicurezza dei dati o le tematiche affrontate nei paragrafi precedenti, ma soprattutto sono da tenere presenti obblighi di conformità a normative o a standard industriali a cui l'azienda utente è sottoposta, in virtù del paese di appartenenza.

Risulta, tuttavia, di fondamentale importanza, la stipulazione e la gestione del contratto tra il cliente e il fornitore di servizi, specificando sia i ruoli distinti, sia le finalità e le modalità essenziali.

I principali soggetti coinvolti in un servizio Cloud computing sono, principalmente, il *fornitore* (cloud provider) da una parte, e il *cliente* (utente) dall'altra.

In certe situazioni, essi sono entrambi "titolari", ovvero coloro che hanno totale autonomia nel trattamento. Ciò accade quando c'è un certo tipo di libertà decisionale nel definire i caratteri essenziali del trattamento e si avrà un insieme di informazioni, scambiato fra entrambi, che verrà considerato fra autonomi titolari del trattamento. In altri casi, si può avere la situazione in cui si hanno un *titolare* e un *responsabile*, rispettivamente, il cliente e il fornitore. Lo scambio di dati può essere considerato come un flusso di informazioni interno alle modalità del contratto.

Il contratto si basa su accordi specifici e non stabiliti unilateralmente, concedendo al cliente maggiori poteri, ottenendo quindi un proprio controllo sulla gestione dei dati.

Tuttavia, il cliente acquisisce la responsabilità degli aspetti organizzativo-gestionali come, per esempio, proporre e adottare delle misure di sicurezza

logiche (Iaas) o controllare l'applicazione delle misure di sicurezza fisiche da parte del fornitore (responsabile) (SaaS, PaaS), di cui servirebbe accertare sempre gli aspetti di affidabilità e competenza.

Le maggiori problematiche avvengono quando il fornitore di servizi si rivolge ad un terzo soggetto esterno, il quale a sua volta dovrebbe essere sotto il controllo del cliente, al pari del fornitore stesso. Appare, quindi, scontato e necessario procedere ad una corretta individuazione del *titolare* e del *responsabile*, per non incorrere in rilevanti conseguenze in tema di determinazione delle responsabilità giuridiche e delle condizioni di sicurezza del trattamento. Tutto ciò per evitare che un'azienda, per esempio, si trovi ad utilizzare inconsapevolmente servizi Cloud privi di autotutela.

Il cliente deve vincolare contrattualmente il fornitore di servizi, nominato come *responsabile* del trattamento dati, nel caso in cui egli decida di affidare a terzi una parte o la completa fornitura del servizio, comprendendo anche i trattamenti di dati personali.

Questo iter prevede passaggi intermedi:

- il fornitore di servizi in cloud deve far presente, al cliente titolare, i terzi fornitori futuri e dove saranno effettuati i trattamenti di dati;
- i terzi fornitori scelti devono accettare di essere i *responsabili* del trattamento dei dati seguendo la normativa prevista;
- i terzi fornitori devono avere nei confronti del cliente titolare lo stesso comportamento che il primo fornitore teneva nei confronti del cliente.

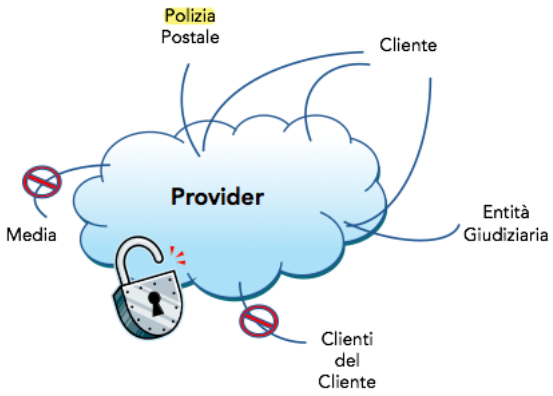

Figura 4.1: Sicurezza

4.2 Trasferimento dati

Attenendosi al Codice della Privacy, riguardo il trasferimento dei dati personali all'estero, ha un aspetto rilevante la localizzazione dei luoghi del trattamento dei dati da parte del fornitore, quando esso si appoggia a luoghi al di fuori dell'Unione europea o a uno stato non membro, che non garantisca un servizio adeguato di sicurezza e protezione.

Il Codice della Privacy vieta a tutti gli effetti il trasferimento dei dati al di fuori dell'Unione Europea, a meno di precise disposizioni.

Infatti, uno dei problemi più evidenti e visibile si riferisce alla tutela contro gli abusi dei dati personali.

Nonostante l'Europa abbia messo in atto una *Direttiva* di protezione dei dati [1], ci sono diverse difficoltà nell'applicazione di tali regole in un ambiente come quello del Cloud computing. La direttiva sulla protezione dei dati ha fornito un certo grado di armonia nell'Unione europea; sussistono però differenze tra le legislazioni nazionali, per esempio, in relazione alle formalità

[1] 95/46/CE: The Data Protection Directive is a European Union directive which regulates the processing of personal data within the European Union. It is an important component of EU privacy and human rights law. On 25 January 2012, the European Commission unveiled a draft European Data Protection Regulation that will supersede the Data Protection Directive.

da compilare a cura del titolare o specifici regimi di protezione dei dati (dati sanitari, dati finanziari, ecc.).

Alcuni aspetti della normativa (in particolare il rigido controllo sulla posizione geografica del trattamento dei dati, come un fattore importante nel determinare le norme applicabili) porta a notevoli difficoltà nella pratica.

Tuttavia, è chiaro che questo non sarà sufficiente ad affrontare tutte le problematiche in questo settore, poichè le norme stabilite dalla Direttiva europea, ovviamente, si applicano solo negli Stati membri, e in un paese non europeo non si è in grado di risolvere i conflitti di legge.

Nell'*art 43 co.1 lett. a,b D.leg. 196/03* si legge che: "Il trasferimento anche temporaneo fuori del territorio dello Stato, con qualsiasi forma o mezzo, di dati personali oggetto di trattamento, se diretto verso un Paese non appartenente all'Unione europea è consentito quando:

- l'interessato ha manifestato il proprio consenso espresso o, se si tratta di dati sensibili, in forma scritta;

- è necessario per l'esecuzione di obblighi derivanti da un contratto del quale è parte l'interessato o per adempiere, prima della conclusione del contratto, a specifiche richieste dell'interessato, ovvero per la conclusione o per l'esecuzione di un contratto stipulato a favore dell'interessato"

La gestione di alcuni server affidata a fornitori in luoghi esterni ai propri, particolarmente in quasi tutti i modelli di servizio Cloud (IaaS, SaaS e PaaS).

Inoltre, questa tipologia si riscontra, specialmente, nel Public Cloud, dove l'infrastruttura è di proprietà di un fornitore che mette i propri sistemi a disposizione di terzi, tramite la rete.

Anche nel Private Cloud, ove l'infrastruttura informatica è per lo più dedicata alle esigenze di una singola organizzazione, il problema sorge quando la gestione è affidata ad un terzo, che ospita il server in una sede esterna alla medesima organizzazione.

In questi casi, per prevenire consistenti problemi, è compito del cliente accertarsi, in relazione al servizio offerto, dove il fornitore ha collocato, "fi-

sicamente", i dati personali, e se questi rimarranno sotto la gestione dello stesso o se verranno trasferiti in terze parti.

Il Cloud deve sempre risiedere in località precise, conosciute dal cliente, rispettando il Codice della Privacy.

Per adempiere alle disposizioni in materia di trasferimento dei dati personali all'estero, il cliente, accertandosi dove sia ubicato il server del fornitore, può assistere a determinate situazioni:

- il server del fornitore è situato nel territorio di uno Stato membro della UE o di uno Stato che garantisca un determinato livello di protezione dei dati personali: in questo caso il trasferimento dei dati verso questi Stati è considerato legale;

- il server del fornitore è situato in uno stato terzo, non giudicato sicuro dalla Commissione UE per garantire un efficiente livello di protezione dei dati: in questo particolare caso, tra il cliente e il fornitore devono essere vigenti clausole contrattuali approvate dalla Commissione UE relative al trasferimento extra UE;

- si considera il fornitore come società dello stesso cliente ed il server si trova in uno Stato non giudicato dalla Commissione UE idoneo nel garantire un buon livello di protezione dei dati personali e sono in vigore le BCR [2]: in tal caso, il trasferimento dei dati personali negli Stati extra UE dove operano le società è giudicato lecito.

  In pratica, l'importanza di questo problema non dovrebbe essere sottovalutata, come la scelta di un sistema giuridico applicabile su base contrattuale è infatti diventata una pratica standard nei contratti di servizi della società dell'informazione.

[2]Binding corporate rules: norme vincolanti di impresa, per consentire a multinazionali, organizzazioni internazionali e gruppi di imprese di fare trasferimenti di dati personali attraverso le frontiere, in conformità con il diritto comunitario sulla protezione dei dati.

Capitolo 5

Service Level Agreement

Il Cloud ha un'architettura diversa in base ai servizi che offre.

I dati sono memorizzati in una posizione centralizzata, chiamata *data center*, avente una grande dimensione di memorizzazione. Di conseguenza le informazioni risiedono, a sua volta, sul server.

I clienti devono avere fiducia nel fornitore, per avere garantita la disponibilità e la sicurezza dei dati.

ll *Service Level Agreement (SLA)* è l'unico accordo legale tra il fornitore del servizio e il cliente.

5.1 Definizione di Service

Questa è la parte più critica dell'accordo tra il *Serivce Provider* (fornitore di servizi) e il *Customer* (cliente), ovvero dove vengono descritti i servizi e il modo in cui questi devono essere proposti. In particolare, include riferimenti alle operazioni del servizio, nonchè le informazioni necessarie per definire le garanzie di livello, nella parte successiva della SLA.

Le informazioni relative al Service devono essere precise e contengono specifiche dettagliate di ciò che esattamente dovrà essere offerto al Customer.

5.2 Definizione di Service Level Agreement

"(...) a formal negotiated agreement between two parties. It is a contract that exists between the Service Provider (SP) and the Customer. It is designed to create a common understanding about service quality, priorities, responsibilities, ecc." [1]

Il *Service Level Agreement (SLA)* è un accordo, tra un *Service Provider* e un *Customer*, che descrive il servizio, gli obiettivi del livello di servizio e specifica le rispettive responsabilità degli attori.

Questo è un elemento estremamente importante di documentazione per entrambe le parti.

Inoltre, gli SLA descrivono gli IT Services, i relativi documenti *Service Level Targets*, specificando le responsabilità dell'IT Service Provider e del Customer.

I Service Level Agreement (SLA) sono sempre più comuni, come un modo per garantire la qualità del servizio, (QoS)[2] nell'informazione, nella comunicazione e nel trasporto/logistica dei servizi, ma sono anche comunemente usati per impostare le condizioni commerciali e di business di una fornitura di servizi (ad es. nel settore delle telecomunicazioni).

Essi non sostituiscono però i contratti formali, soggetti al controllo giurisdizionale, ma permettono comunque di ridurre il numero complessivo e il costo di tali contratti.

Se usato correttamente dovrebbe:

- identificare e definire le esigenze del Customer;
- fornire un quadro di riferimento per la comprensione, semplificando questioni complesse;
- ridurre i conflitti tra le parti;

[1] "Service Level Agreement / Quality of Service Overview", TM Forum. [Online].

[2] È usato per indicare i parametri usati per caratterizzare la qualità del servizio offerto dalla rete (ad esempio perdita di pacchetti, ritardo), o gli strumenti o tecniche per ottenere una qualità di servizio desiderata

- dare la possibilità al Customer di ottenere quello che effettivamente pretende dal servizio.

5.3 Caratteristiche principali SLA

5.3.1 Performance Management

La natura dinamica del Cloud computing richiede un continuo monitoraggio dei requisiti per far rispettare gli SLA. I Customers potrebbero non fidarsi completamente delle misure prese da un fornitore di servizi, i quali potrebbero richiedere l'intervento di un terzo, per avere una misurazione critica dei parametri di servizio.

Non a caso uno SLA è un insieme di *Service Level Objective* che sono indicatori di performance. Un gruppo appropriato di servizi concreti/risorse può essere scelto in modo da garantire il rispetto degli SLA e soddisfare i consumatori.

Service Level Objective:

- obiettivo di qualità di servizio che deve essere raggiunto;
- set di misure KPI [3], con soglie per decidere se l'obiettivo è soddisfatto o meno.

Quindi, è di fondamentale importanza, in un contratto, monitorare e misurare il livello di prestazioni del servizio.

Essenzialmente, ogni servizio deve poter essere misurato ed i rispettivi risultati, analizzati e riportati.

I parametri di riferimento, obiettivi e metriche che vengono utilizzati, devono essere specificati nel contratto stesso.

Il livello di prestazioni di servizio deve essere rivisto regolarmente dalle due parti.

[3]Un indicatore chiave di prestazione (Key Performance Indicators o KPI) è un indice che monitora l'andamento di un processo aziendale.

5.3.2 Metrics

I parametri della SLA sono specificati dalle metriche di sicurezza. Quest'ultime sono le definizioni di valori di proprietà del servizio che sono misurate da un sistema che fornisce altri parametri e costanti.

Le metriche sono lo strumento chiave per descrivere esattamente ciò che intendiamo per *SLA Parameters* specificando come misurare o calcolare i valori dei parametri. Le specifiche dei parametri del livello di servizio e le metriche rappresentano anche la comprensione comune delle parti.

5.3.3 Quality of Service

Come accennato precedentemente, lo SLA si basa sul Quality of Service, che è definito come: *"the collective effect of service performances, which determine the degree of satisfaction of a user of the service. The quality of service is characterized by the combined aspects of service support performance, service operability performance, service integrity and other factors specific to each service."* [4].

Per ottenere una buona descrizione di SLA, consiste analizzare vari aspetti:

- descrizione della natura del servizio fornito: comprende il tipo del servizio, la descrizioni dei problemi tecnici connessi con il servizio, come ad esempio, la connettività di rete, il funzionamento e la manutenzione. Inoltre, anche le configurazioni del server e del client;
- il livello di risposta e affidabilità del servizio: include requisiti di disponibilità e in quanto tempo il servizio si esibisce in uno stato normale;
- servizio di segnalazione del problema: questo implica la notifica in caso di segnalazioni di errori, con annessi strumenti per la rapida risoluzione;
- tempi di risposta e risoluzioni problemi: questo definisce il tempo dopo il quale i problemi devono essere risolti;

[4] International Telecommunications Union(ITU-T)

- monitoraggio e servizio di segnalazione: descrive come devono essere i livelli di qualità di monitoraggio e comunicazione, e con quale frequenza devono essere utilizzati;
- debiti sul fornitore del servizio, se le promesse non sono state soddisfatte: in tali casi, a un cliente di servizio gli può essere dato credito extra, come il cliente può decidere di risolvere il contratto, o chiedere un rimborso;
- condizioni supplementari: si tratta di condizioni in cui il contratto di servizio non è più valido. Questo può essere riscontrato, per esempio, nel caso di cause naturali (ad esempio inondazioni, incendi, ...).

5.3.4 Creazione di SLA

La creazione di una SLA consiste in una serie di passaggi:

1. *SLA development*: sviluppo di modelli e diritti;
2. *negotiation and Sales*: negoziare ed eseguire i contratti;
3. *implementation*: generare, fornire il servizio e monitorarlo;
4. *execution*: funzionamento e mantenimento, monitorando le prestazioni;
5. *assessment*: valutare le prestazioni e considerare nuovi modelli.

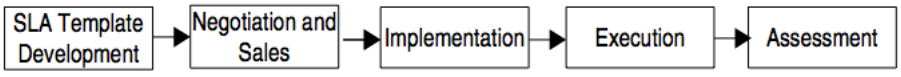

Figura 5.1: SLA life cycle

5.4 Struttura SLA

La struttura generale di tutti i diversi SLA rimane la stessa:

- le parti coinvolte (Provider e Customer);
- SLA parameters;
- le metriche sfruttate per computare gli SLA parameters;
- gli algoritmi inerenti;
- i *Service Level Objectives* (SLO) e le contro misure da intraprendere nel caso vengano violati gli SLOs

5.5 Approcci generali alla SLA

Generalmente, per approcciarsi alla SLA, ci si rifà a tre aspetti principali: *Insurance, Provisioning, Adaptive*.

Il primo, *Insurance*, è considerato il più importante, in quanto individua e monitora quelli che sono gli obiettivi del servizio offerto (es: performance, disponibilità, tempi di risposta). Inoltre, rilascia una relazione SLA, ovvero vengono inclusi incontri periodici con i contraenti del contratto, per discutere dello stato di conformità degli SLA.

Il secondo aspetto, *Provisioning*, utilizza tecniche di configurazione per supportare gli SLA all'interno della rete. In particolare individua gli obiettivi del servizio da fornire a ciascun cliente, determina la configurazione del sistema da utilizzare per ciascuno dei clienti.

Per ultimo, l'*Adaptive*, aggiunge un altro aspetto che si adatta al *Provisioning*: in aggiunta rispetto agli altri punti, controlla che se il monitoraggio avvisi di una possibile violazione degli obiettivi, riadatta la configurazione del cliente per garantire al meglio gli obiettivi del servizio.

5.6 SLA: lato client e lato provider

5.6.1 Lato client

L'obiettivo principale dei fornitori di SaaS è quello di minimizzare i costi e di massimizzare il livello di soddisfazione del cliente. Quest'ultima dipende in che misura è soddisfatto lo SLA.

Quando un client fa una richiesta a livello applicativo, questa viene inviata al provider SaaS con vincoli di QoS, come ad esempio:

- *deadline*: tempo massimo che l'utilizzatore vorrebbe attendere;
- *budget*: quanto un utente è disposto a pagare per ottenere i servizi richiesti;
- *penalty rate ratioe*: si intende un risarcimento dei consumatori se il provider SaaS non rispetta la scadenza;
- *input file size*: dimensione dei file in input forniti dagli utenti;
- *request length*: quante istruzioni devono essere eseguite per rispondere alla richiesta;

5.6.2 Lato provider

- **SaaS provider**

  Un provider SaaS utilizza risorse da fornitori IaaS e software leasing come servizi per utenti.

  È importante stabilire due "rapporti" nello SLA, perchè lo SLA con l'utente può aiutare il provider SaaS, per migliorare il livello di soddisfazione dei cliente, guadagnando la fiducia degli utenti con la qualità del servizio; lo SLA con i fornitori di risorse, può servire per soddisfare il servizio.

  Se una parte del contratto viola i termini, l'inadempiente deve pagare la sanzione in base alle clausole definite nella SLA.

- **IaaS provider**

  Un provider IaaS, offre VM fornitori ai SaaS provider ed è responsabile del funzionamento delle VM sulle proprie risorse fisiche.

  SLA provider comprende le seguenti caratteristiche:

  - *Service Initiation Time*: quanto tempo occorre per installare un servizio;
  - *Price*: quanto costa ad un SaaS provider usufruire di risorse del IaaS provider;
  - *Input Data Transfer Price*: quanto costa ad un SaaS provider trasferire dati da risorse locali a risorse virtuali (VM);
  - *Output Data Transfer Price*: quanto costa ad un SaaS provider trasferire dati da risorse virtuali a risorse locali;
  - *Processing Speed*: velocità di elaborazione delle risorse;
  - *Data Transfer Speed*: velocità di trasferimento dati, dipendente soprattutto dalla velocità di rete;

5.7 Problematiche SLA

5.7.1 Problem Management

Lo scopo della gestione dei problemi è quello di minimizzare l'impatto negativo con i rischi e i problemi.

Questo specifica che di solito ci deve essere un adeguato processo per gestire e risolvere gli imprevisti e che ci deve essere anche l'attività di prevenzione.

5.7.2 Customer Duties and Responsibilities

È importante che il Customer capisca che possiede anche la responsabilità di sostenere il processo di erogazione del servizio.

In genere, il Customer deve provvedere all'organizzazione di accesso, alle strutture e alle risorse per i dipendenti del fornitore che ne hanno bisogno.

5.7.3 Monitoring di SLA

Per avere un controllo, continuamente, sui valori della qualità del servizio, si necessità di diversi approcci di monitoraggio.

Esistono due principali tipologie di monitoring: *Server-side, Client-side.*

Da un lato, l'approccio Server-side è più legato all'implementazione, che non sempre può essere accessibile, mentre quando si parla a livello di Client-side, rimane più indipendente dall'implementazione del servizio, ma i valori potrebbero non sempre essere aggiornati.

Sia a livello di Server-side sia Client-side, è possibile avere il controllo di misurazioni come latenza, throughput, tasso di trasferimento, tempo di risposta dei messaggi, disponibilità e affidiabilità. Invece, per misurazioni come scalabilità, robustezza, capacità, gestione delle eccezioni e stabilità vengono ottenute solo dal lato Server-side.

5.8 Sicurezza SLA

5.8.1 Security Service Level Agreement (Sec-SLA)

La sicurezza è un aspetto particolarmente critico di ogni SLA.

Il Customer deve fornire un accesso controllato, fisico e logico, alle proprie informazioni. Allo stesso modo, il fornitore deve rispettare e attenersi alle politiche di sicurezza del Customer e delle varie procedure.

Allora sorge subito una domanda: "come viene accuratamente delineata la sicurezza in questo contesto?". La risposta a questa domanda la si può vedere in tre punti principali:

- *Policy analysis*: tutti i dati possibili possono essere valutati in questa fase, per creare una Sec-SLA. Viene analizzata la documentazione disponibile nel campo in cui l'organizzazione ha le sue attività, come i contratti dei clienti, i regolamenti nazionali, le politiche interne e così via;

- *Architecture analysis*: l'obiettivo è quello di analizzare l'infrastruttura del cliente e di trovare i requisiti che potrebbero essere risolti direttamente in elementi come i server web e firewall;

- *Interviews*: si raccolgono informazioni sui problemi di sicurezza dal punto di vista dell'utente.

5.8.2 Cos'è il Sec-SLA

Security Service Level Agreements (Sec-SLA), non è una nuova tecnica, ma più come uno nuovo "design" per gli accordi di livello di servizio.

Sec-SLA è una specifica SLA che offre metriche relative alla sicurezza, invece dei tradizionali aspetti riguardo le telecomunicazioni, come per esempio, throughput, ritardi, pacchetti persi.

I requisiti di sicurezza a livello di servizio o richieste vengono convertiti in una serie di meccanismi, tra cui la crittografia, i dati packet filtering [5], la ridondanza di hardware e software, ecc.

Oltre all'introduzione di un Sec-SLA, di rilevante importanza è il monitoraggio dello stesso. Quest'ultimo è anche molto complesso poichè l'approccio che l'agente-manager ha per altri tipi di SLA, non è adattabile ad ogni caratteristica di sicurezza.

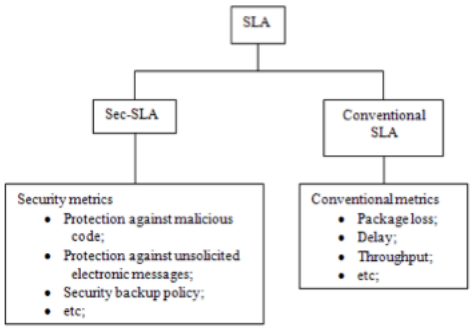

Figura 5.2: Sec-SLA Conventional SLA

La principale caratteristica di un Sec-SLA è quella di cercare di formalizzare metriche di sicurezza. L'obiettivo è quello di creare metriche il più possibili quantificabili e misurabili.

In molti casi si è cercato di attenersi, il più possibile, a norme internazionali relative a condizioni di sicurezza dell'informazione.

Alcuni esempi di metriche possono essere:

- *Password management*: ogni quanto dovrebbe essere cambiata una password;
- *Backup policies*: ogni quanto viene effettuato un backup, e le modalità con cui viene realizzato;
- *Repair time*: quanto tempo serve per i team IT per mettere un sistema in stato di *failure*.

[5] È il più semplice firewall, che si limita a valutare gli header di ciascun pacchetto, decidendo quali far passare e quali no sulla base delle regole configurate.

5.9 Linguaggio di programmazione per SLA

Appropriati linguaggi formali basati su XML sono stati progettati per consentire la definizione di SLA in tutti i suoi elementi.

In particolare si ha il *documento XML* più *XML schema-based*.

Il primo include le informazioni del messaggio, mentre lo schema XML può essere utilizzato per definire la struttura dei messaggi e i tipi di dati secondo lo SLA.

Così siamo in grado di utilizzare lo schema XML per convalidare i documenti XML e renderli conformi alle specifiche dello schema.

Il primo obiettivo di un linguaggio che definisce uno SLA è di fornire la capacità di esprimere, con il massimo grado di precisione, le caratteristiche qualitative e quantitative di un servizio.

Grazie a questo tipo di strumento, le parti sono in grado di poter accordarsi in modo preciso e dettagliato sul livello di un particolare servizio.

Altri risultati rilevanti sono la possibilità di far riferimento ad uno standard, che tutti sono in grado di comprendere ed utilizzare, realizzare in maniera semplice il confronto fra le diverse offerte, ragionare sulle proposte del servizio, in modo da capire cosa si possa offrire e ricevere ed infine controllare facilmente le garanzie della qualità del servizio.

Alcuni principali requisiti:

- *Parametrizzazione*: ogni SLA include un serie di parametri, che sono i principali elementi di descrizione di un servizio;

- *Composizione*: i fornitori del servizio devono essere in grado di comporre SLA che possano offrire nuovi servizi ai clienti, cercando anche di offrire un servizio di cooperazione tra le diverse entità del dominio;

- *Validazione*: uno SLA, deve essere convalidata dalle parti contraenti, prima di essere iniziata;

- *Monitoraggio*: le rispettive parti dovrebbero, costantemente, controllare che i livelli del servizio stabiliti, vengano correttamente forniti e se sono stati soddisfatti dai propri clienti.

5.10 XML e SLA

Di seguito, verrà descritta la struttura generale dei messaggi di SLA per i diversi servizi internet. Nella maggior parte dei casi, SLA si compone di tre parti:

- una descrizione delle parti e dei loro ruoli (fornitore, cliente, terzi);
- una descrizione dettagliata dei parametri del livello di servizio;
- una rappresentazione degli obblighi delle parti.

5.10.1 Rappresentazione delle parti in XML

Questa implementazione propone le verie informazioni sul fornitore di servizi, il cliente e dei terzi.

I terzi, nominati come *Supporting Parties*, sono coinvolti nella misurazione dei parametri del servizio, controllano le garanzie rilasciate e la gestione delle procedure di correzione, da attuare in caso di guasto.

```
<xsd:complexType name=PartiesType''>
<xsd:sequence>
   <xsd:element name=''ServiceProvider''  type=''sla:PartyType''/>
   <xsd:element name=''ServiceConsumer''  type=''sla:PartyType''
   <xsd:element name=''SupportingParty''  type=''sla:PartyType''minOccurs=''0''
                                     minOccurs='0'
                                     maxOccurs=''unbounded''/>
</xsd:sequence>
</xsd:complexType>

</xsd:complexType>
<xsd:complexType nme=''PartyType''>
<xsd:sequence>
          <xsd:element name=''Contact''
type=''sla:ContactInfonnationType''/>
          </xsd:sequence>
          <xsd:attribute name=''name'' type=''xsd:string'' />
</xsd:complexType>
```

5.10.2 Descrizione del servizio in XML

In questa sezione, si mettono in evidenza per quale servizio vengono usati i parametri, quali e come sono usati.

```
<xsd:complexTypename=''Servicetype''>
  <xsd:complexContent>
    <xsd:sequence>
      <xsd:element name=''ServType''    type=''xsd:string''
                                maxOccurs=''1''/>
      <xsd:element name=''ServiceLevel''     type=''xsd:string''
                                maxOccurs=''1''/>
        <xsd: element name=''ServiceMetric'' type=''sla:MetricType''
                                maxOccurs=''unbounded''/>
    <xsd:sequence>
  <xsd:complexContent>
<xsd:complexType>
```

5.10.3 Descrizione metrica del servizio in XML

La descrizione della metrica implica un'analisi dei parametri usati, per valutare il livello del servizio.

```
<xsd:complexTypename=''ServiceMetric''>
<xsd:complexContent>
   <xsd:sequence>
      <xsd: element name=''Expression''
         type=''sla:LogicExpressType''
         maxOccurs=''unbounded''/>
        <xsd:element name=''MetricGrade''
            type=''xsd:integer'' MaxOccurs=''unbounded''/>
   </xsd:sequence>
 <xsd:complexContent>
<xsd:complextype>

<ServiceDefinition name=''Downloadservice''>
  <ServiceDescription name=''Download''>
      <ServType> Download</ServType>
  <ServiceLevel> bronze ServiceLevel>
  </ServiceDescription>
<ServiceMetric>
<Expression>
<Predicate xsi: type=''sla:less''>
<SLAParameter>AverageResponseTime
</SLAParameter>
          <Value>5</Value>
          <Predicate>
 </Expression>
<MetricGrade> 4 <MetricGrade>
</ServiceMetric>
<ServiceMetric>
```

5.10.4 Descrizione obblighi in XML

Gli obblighi rappresentano i rispettivi requisiti che le parti si impegnano a rispettare.

```
<xsd:complexType name=''ServiceLevelObjectiveType''>
<xsd:sequence>
        <xsd:element name=''Obliged'' type=''xsd:string''/>
        <xsd:element name=''validity''
type=''sla:periodType'' maxOccurs=''unbounded''/>
        <xsd:element name=''action''
type=''sla:actionType''/>
</xsd:sequence>
</xsd:complexType>
```

Il periodo di validità è specificato, indica gli intervalli di tempo per il quale un parametro SLA dato è valido.

Le azioni vengono eseguite ogni volta che una violazione di un obiettivo di livello di servizio si è verificato.

5.10.5 Descrizione azione in XML

```
<xsd: ComplexType name=''actionType''>
<xsd:sequence>
<xsd:element name=''Expression''
type=''sla:LogicExpressionType''/>
<xsd:element ref=''sla:QualifedAction''
            maxOccurs=''unbounded''/>
</xsd:sequence>
<xsd:attribute name=''name'' type=''xsd.string''/>
</xsd:complexType>
```

L'espressione *LogicExpressionType* definisce un requisito dell'azione.

Il *QualifiedAction* contiene una definizione delle azioni da eseguire.

5.11 SLA per il Cloud Computing

La tendenza del Cloud Computing è vista come un'estensione del paradigma SOA.

Si cerca di tutelare, nel migliore dei modi, il processo di gestione dell'accordo provider-consumer.

Con *Web Service Level Agreement* si propone un'architettura per la gestione dei consumatori Cloud, fornendo SLA.

5.11.1 Web Service Level Agreement

Diverse specifiche per definire SLA sono state proposte per i servizi web.

Il linguaggio *Web Service Level Agreement* introduce un meccanismo per aiutare gli utenti di servizi web a configurare e controllare le loro risorse, al fine di soddisfare il livello di servizio.

Inoltre, gli utenti sono in grado di monitorare i parametri di SLA durante la fase di esecuzione e riferiscono notifiche in caso ci siano violazioni.

WSLA, è stato sviluppato per descrivere i servizi in tre categorie:

1. *parti*: in questa sezione, si hanno informazioni sui consumatori dei servizi, sui fornitori di servizi e su terzi;

2. *parametri SLA*: qui vengono descritti i principali parametri che sono suddivisi in due tipi di metriche;

   (a) *metriche delle risorse*: le risorse vengono recuperate direttamente dal provider e sono utilizzate così come sono, senza ulteriori elaborazioni;

   (b) *metriche composte*: questa metrica viene utilizzata per rappresentare il calcolo della combinazione di informazioni su un fornitore, una combinazione di metriche di risorse diverse, calcolate secondo un algoritmo specifico;

3. *Service Level Objective (SLO)*: in questo ambito, vengono messi in evidenza gli obblighi e le azioni che vi sono tra il fornitore di servizi e il cliente, dei vari servizi web. Inoltre sono definiti i concetti di gestione del servizio, costi e altri obiettivi di servizi.

   Essi possono essere considerati come espressioni formali, della struttura conosciuta *if...then*: nell'*if* si ha la *condizione*, mentre nella conseguenza (*then*) si ha l'*azione*.

WSLA garantisce un livello di monitoraggio, ma non definisce in modo chiaro quale livello di servizio può essere considerato di violazione.

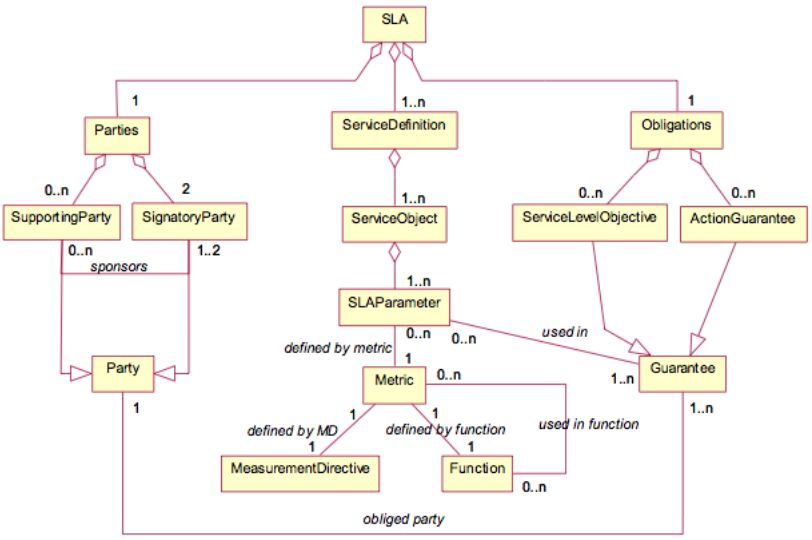

Figura 5.3: Main concepts of WSLA

5.11.2 Servizi WSLA

La nube è intrinsecamente dinamica e l'utilizzo delle risorse cambia continuamente.

Quindi si adatta perfettamente con SLA, che hanno bisogno di abbracciare questa natura dinamica.

Inoltre, a causa delle crescenti preoccupazioni di privacy e sicurezza dei dati, i consumatori potrebbero divulgare alcuni dettagli al Cloud provider.

Una volta che il documento SLA è stabilito, deve essere distribuito. La distribuzione SLA è definita come il processo di convalida per le parti coinvolte.

Descriviamo tre servizi comuni WSLA , e alcuni dei loro adeguamenti richiesti nel contesto Cloud.

1. *Measurement Services*: questi servizi sono responsabili di misurare dei parametri a "runtime" relativi alle risorse del Cloud provider. Come discusso in precedenza, i parametri di servizio, come il tempo di risposta e il throughput vengono costantemente aggiornati a causa della variabilità della richiesta di servizio da parte del consumatore.

   Nel contesto del Cloud Computing, tuttavia, i parametri di utilizzo e di costo sono dinamici. Ciò è dovuto al "pay-as-you-go", che caratterizza la natura e l'elasticità del Cloud.

   Si possono identificare servizi aggiuntivi al Cloud Computing, come l'utilizzo, il costo/prezzo dei dati importanti.

2. *Condition Evaluation Service*: questo servizio è responsabile di ottenere i risultati di altri servizi, riguardo la misurazione e la valutazione degli obiettivi di livello.

   Se ci sono violazioni del servizio di gestione verrà contattato il *Management service*.

A causa della natura dinamica del Cloud, la valutazione delle condizioni deve essere eseguita più frequentemente, rispetto a quello che avviene in un contesto tradizionale.

3. *Management Service*: questo servizio è incaricato di controllare violazioni dei Service Level Objectives.

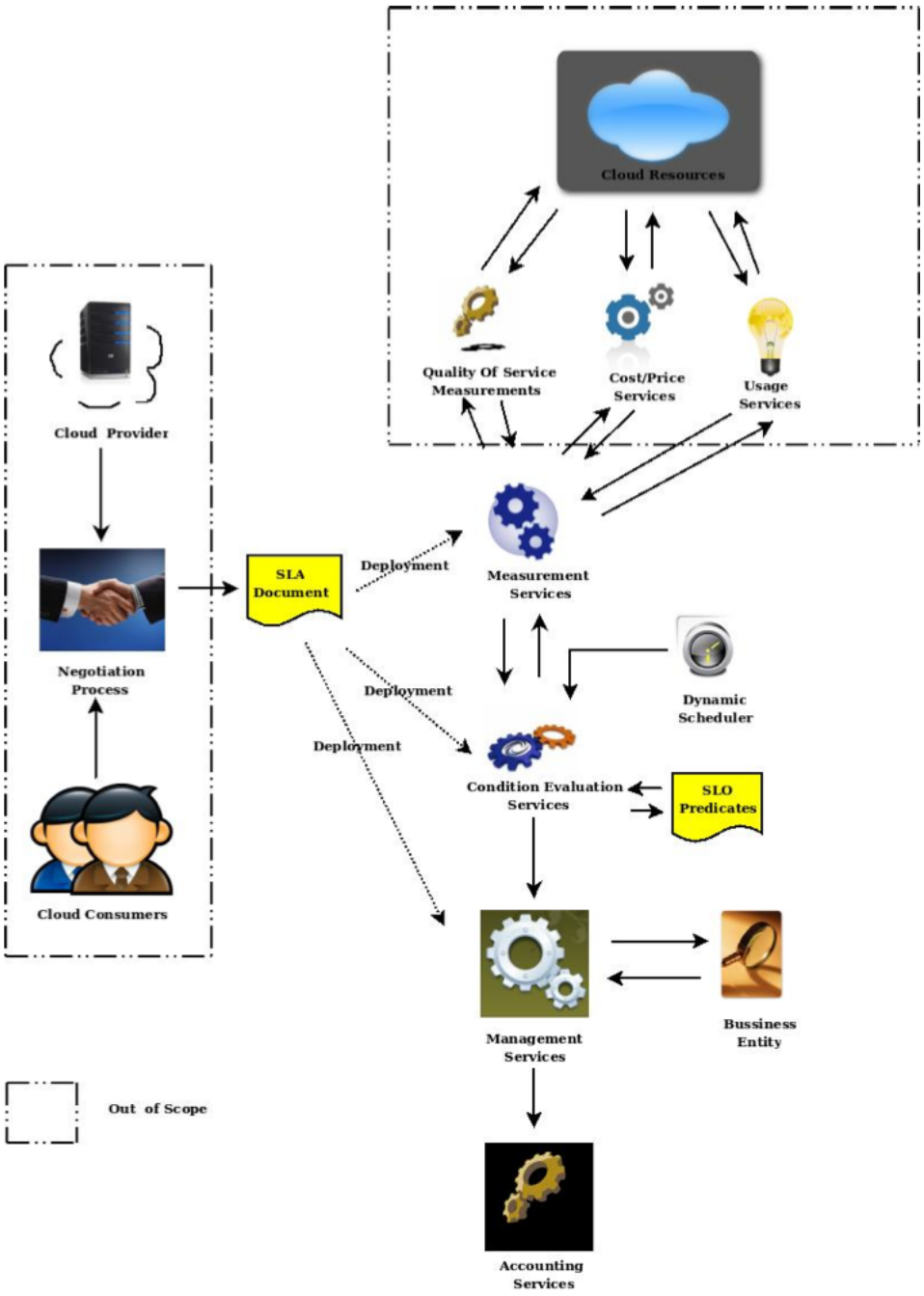

Figura 5.4: Architecture WSLA

5.12 Ajaxplorer

Ajaxplorer è un file manager via web, open source, basato su un'infrastruttura Cloud (IaaS), il quale dà la possibilità di condividere un gran numero di file, da singoli, organizzazioni, aziende.

Ajaxplorer è realizzato maggiormente in PHP, per quanto riguarda il lato server, mentre JAVASCRIPT per gli aspetti inerenti al client-side. Inoltre, XML ha un ruolo importante per la gestione dei dati e metadata usati nel software, DOM e XPATH sono molto utilizzati sia su lato server sia su lato client per navigare attraverso i dati XML.

Le principali funzionalità sono utilizzate come "plug-in", in modo da permettere di introdurre funzionalità aggiuntive, overriding e creare dipendenze tra esse.

Ajaxplorer sfrutta il collegamento con un CMS, *Content Management System*, che si appoggia a sua volta su un server già esistente, che permette, in questo specifico caso, la gestione del meccanismo di autenticazione dello user.

Analizzando i sorgenti del software, si possono notare alcuni esempi di come vengono gestiti aspetti di sicurezza, a livello di implementazione.

5.12.1 Alcuni aspetti di sicurezza in Ajaxplorer

Innanzitutto, si può capire come è gestita l'autenticazione di uno user, nel momento in cui accede. Nei repository di Ajaxplorer si può notare come vengano gestite e nascoste le credenziali dei singoli user. Nei seguenti esempi implementativi, si hanno due funzioni di codifica e decodifica della password appartenente ad uno user, che una volta salvata nel server viene criptata o decripatata attraverso la funzione *MCrypt*.

```
class AJXP_Safe{

private static $istance;

private $user;
private $econdedPassword;
private $secretKey;
private $separator = ''__SAFE_SEPARATOR__'';
private $forceSessionCredentials = false;

public function __construct(){
if(defined('AJXP_SAFE_SECRET_KEY')){
                $this->secretKey = AJXP_SAFE_SECRET_KEY;
        }else{
                $this->secretKey = ''\1CDAFxo''p#'';
        }
}
```

Funzione che codifica la password.

```
private function _encodePassword($password, $user){
if (function_exists('mcrypt_encrypt')){
          $iv = mcrypt_create_iv(mcrypt_get_iv_size(MCRYPT_RIJNDAEL_256,
                  MCRYPT_MODE_ECB), MCRYPT_RAND);
          $password = base64_encode(mcrypt_encrypt(MCRYPT_RIJNDAEL_256,
                  md5($user.$this->secretKey), $password, MCRYPT_MODE_ECB, $iv));
           }
return $password;
}
```

Funzione che decodifica la password.

```
private function _decodePassword($encoded, $user){
if (function_exists('mcrypt_decrypt')){
        $iv = mcrypt_create_iv(mcrypt_get_iv_size(MCRYPT_RIJNDAEL_256,
                        MCRYPT_MODE_ECB), MCRYPT_RAND);
        $encoded = trim(mcrypt_decrypt(MCRYPT_RIJNDAEL_256,
                        md5($user.$this->secretKey), base64_decode($encoded),
                        MCRYPT_MODE_ECB, $iv));
}
return $encoded;
}
```

```
Repository: ajaxplorer-core-4.2.2.core.classes.class.AJXPSafe.php
```

Inoltre, Ajaxplorer sfrutta il cosiddetto "WebDAV", *Web-based Distributed Authoring and Versioning*, il quale dà a disposizione un set di istruzioni che permettono di gestire file in un server remoto, accedendo da vari tipi di client.

In questa sezione di implementazione, si può notare come vengano riutilizzate le funzioni precedentemente descritte, ma in aggiunta si ha il *Token Pattern*.

Quest'ultimo viene visto come possibile tecnica per cercare di evitare e tenere sotto controllo il "CSRF", *Cross-site request forgery*.

Assegnazione di un lock-token ad uno user.

```
public function assignLock( $user, $lockToken ){
        AJXP_Logger::debug("ASSIGNING DAVLOCK $user $lockToken");
        if(!isSet($this->tokens[$user])){
            $this->tokens[$user] = array();
        }
        $this->tokens[$user][$lockToken] = true;
    }
```

Si controlla che il lock sia stato assegnato o no allo user.

```
public function ownsLock( $user, $lockToken ){
        AJXP_Logger::debug("TESTING DAVLOCK $user $lockToken");
        return (isset($this->tokens[$user][$lockToken]));
    }
```

Il lock viene rilasciato. In questo caso *ownsLock* tornerà false, poichè il lock non è più in possesso dello user.

```
public function releaseLock( $user, $lockToken ){
        AJXP_Logger::debug("RELEASING DAVLOCK $user $lockToken");
        unset($this->tokens[$user][$lockToken]);
    }


}
...}
Repository: ajaxplorer-core-4.2.2.core.classes.class.AJXP_WebdavAuth.php
```

Un altro esempio di uso di questo pattern lo si nota nella gestione dell'autenticazione di un utente.

```
static function generateSecureToken(){
$_SESSION["SECURE_TOKEN"] = md5(time());
return $_SESSION["SECURE_TOKEN"];
}

static function getSecureToken(){
return (isSet($_SESSION["SECURE_TOKEN"])?$_SESSION["SECURE_TOKEN"]:FALSE);
}

static function checkSecureToken($token){
if(isSet($_SESSION["SECURE_TOKEN"]) && $_SESSION["SECURE_TOKEN"] == $token){
return true;
}
```

```
return false;
}
Repository: ajaxplorer-core-4.2.2.core.classes.class.AuthService.php
```

Un modello di prevenzione ai problemi di sicurezza, è il controllo di minacce come "Brute Force Attack", e di conseguenza l'eliminazione di tentativi di "Denial of Service", sfruttando come risoluzione primaria "Captcha". Brute Force Attack consiste nel tentare di attaccare tutti i dati crittografati di un sistema. Captcha è un tipo di *challenge-response* usato per assicurarsi che il tentativo di accesso sia effettuato realmente da un essere umano. Nello specifico cerca di evitare l'esecuzione di BOT, ovvero script creati per emulare il comportamento di una persona fisica nell'esecuzione di determinati compiti.

Infatti, quando un computer passa sotto il controllo di uno BOT può essere utilizzato per svolgere alcune prestazioni automatiche, tra queste Denial of Service.

Come già mostrato nei capitoli precedenti, con il DoS si intende un attacco sulle prestazioni di un qualsiasi sistema web, fino a portarlo al limite delle prestazioni.

In questa funzione si controlla quante volte viene tentato l'accesso da parte di un utente.

```
 static function checkBruteForceLogin(&$loginArray)
    {
     $serverAddress = "";
     if(isSet($_SERVER['REMOTE_ADDR'])){
     $serverAddress = $_SERVER['REMOTE_ADDR'];
     }else{
     $serverAddress = $_SERVER['SERVER_ADDR'];
     }
     $login = null;
     if(isSet($loginArray[$serverAddress])){
        $login = $loginArray[$serverAddress];
     }
        if (is_array($login)){
            $login["count"]++;
        } else $login = array("count"=>1, "time"=>time());
        $loginArray[$serverAddress] = $login;
        if ($login["count"] > 3) {
            if(AJXP_SERVER_DEBUG){
                AJXP_Logger::debug("DEBUG : IGNORING BRUTE FORCE ATTEMPTS!");
                return true;
            }
            return FALSE;
        }
        return TRUE;
    }
```

Repository: ajaxplorer-core-4.2.2.core.classes.class.AuthService.php

In questa sezione, è messa in luce l'implementazione di *CaptchaProvider.php*. Essa si appoggia alla classe *securimage.php*, che serve per la generazione e autenticazione di immagini.

```
class CaptchaProvider{
public static function sendCaptcha(){

$libPath = AJXP_BIN_FOLDER."/securimage";

$img = new Securimage();
$img->wordlist_file = $libPath."/words/words.txt";
$img->gd_font_file = $libPath."/gdfonts/automatic.gdf";
$img->signature_font = $img->ttf_file = $libPath."/AHGBold.ttf";

$img->image_height = 80;
$img->image_width = 170;
$img->perturbation = 0.85;
$img->image_bg_color = new Securimage_Color("#f6f6f6");
$img->multi_text_color = array(new Securimage_Color("#3399ff"),
                               new Securimage_Color("#3300cc"),
                               new Securimage_Color("#3333cc"),
                               new Securimage_Color("#6666ff"),
                               new Securimage_Color("#99cccc")
                               );
$img->use_multi_text = true;
$img->text_angle_minimum = -5;
$img->text_angle_maximum = 5;
$img->use_transparent_text = true;
$img->text_transparency_percentage = 30; // 100 = completely transparent
$img->num_lines = 5;
$img->line_color = new Securimage_Color("#eaeaea");
$img->signature_color = new Securimage_Color(rand(0, 64),
```

```
rand(64, 128), rand(128, 255));
$img->use_wordlist = true;
if(!function_exists('imagettftext')){
$img->use_gd_font = true;
$img->use_transparent_text = false;
$img->use_multi_text = false;
}
//$img->show($libPath."/backgrounds/bg3.jpg");
$img->show();
}
Repository: ajaxplorer-core-4.2.2.core.classes.class.CaptchaProvider.php
```

Un altro aspetto fondamentale, nella gestione della sicurezza a livello di trasferimento dati, si ha nell'uso del protocollo a crittografia asimmetrica, ovvero HTTPS, *HyperText Transfer Protocol over Secure Socket Layer*.

Esso risulta indispensabile per garantire trasferimenti sicuri di dati attraverso il web, senza rischiare intercettazioni da parte di terzi.

Conclusioni

Con questa tesi si è cercato di dare una panoramica sugli aspetti principali di sicurezza nel Cloud computing.

Innanzitutto, si è mostrato come il Cloud computing sia nato, si sia sviluppato e come velocemente stia diventando una delle tecnologie più diffuse e più sfruttate.

Si è analizzato come in questo ambito siano presenti diversi aspetti problematici sia riguardo le classiche difficoltà di sicurezza, sia gli aspetti giuridici e legali.

Si parla di rischi come perdita di informazioni personali, credenziali, dati sensibili, fino a riconoscere violazioni di privacy riguardanti un singolo individuo; alcune tecniche risolvono diversi problemi, lasciando però aperte molte altre questioni.

Inoltre, sono stati messi in rilievo alcuni punti di normative che regolano la gestione e il trasferimento di un certo tipo di dati da parte del fornitore di servizi Cloud. In alcuni casi, i dati inconsapevolmente vengono gestiti senza attenersi a regole esistenti, come per esempio il trasferimento dei dati all'estero.

Un aspetto rilevante su cui si è posta molta attenzione è il rapporto tra il Customer e il Provider, mostrando gli specifici ruoli e le rispettive competenze, ponendoli nell'ambito principale in cui essi operano: il Service Level Agreement. Esso è l'unico accordo legale che regola il rapporto tra i due attori principali, per definire i termini di servizio richiesto,per le garanzie opportune da ottenere e soprattutto per la validità dei principali punti del

contratto stesso. Partendo dalla creazione di questo, si arriva allo sviluppo e per finire all'implementazione in XML dei termini specificati.

Se da una parte il Cloud offre numerosi vantaggi a livello di servizi, allo stesso tempo, però, può risultare un problema non porre attenzione agli aspetti di sicurezza, come in qualsiasi altra tecnologia informatica.

Naturalmente, grazie alla continua evoluzione del mondo tecnologico, si dovrebbe arrivare a più chiare e dettagliate soluzioni.

Bibliografia

[1] Wikipedia - http://en.wikipedia.org

[2] IEEE Explore - http://ieeexplore.ieee.org

[3] SSRN - http://www.ssrn.com/

[4] https://privacycloudmobile.clusit.it/

[5] CSA - https://cloudsecurityalliance.org/

[6] Balanchandra Reddy Kandukuri, Ramakrishna Paturi V, Atanu Rakshit, *Cloud Security Issues*, IEEE, 2009

[7] Neil Robinson, Lorenzo Valeri, Jonathan Cave, *The Cloud: Understanding the Security, Privacy and Trust Challenges*, SSRN, 2010

[8] Shirlei Aparecida de Chaves, Carlos Becker Westphall, Flavio Rodrigo Lamin, *SLA Perspective in Security Management for Cloud Computing*, IEEE, 2010

[9] Richard Mosher, *Cloud Computing Risks*, Usa, ISSA, 2011

[10] Kresimir Popovic, Zeljko Hocenski, *Cloud computing security issues chalenges*, Croazia, IEEE, 2010

[11] Saleem-ullah Lar, Viaofeng Liao, Syed Ali Abbas, *Cloud Computing Privacy Security. Global Issues, Challenges, Mechanisms*, China, IEEE, 2011

[12] Massimiliano Rak, Loredana Liccardo, Rocco Aversa, *A SLA-based interface for security management in Cloud and GRID Integrations*, IEEE, 2011

[13] Jean-Henry Morin, Jocelyn Aubert, Benjamin Gateau, *Towards Cloud Computing SLA Risk Management: Issues and Challenges*, Hawaii, IEEE, 2012

[14] Pankesh Patel, Ajith Ranababu, Amit Shath, *Service Level Agreement in Cloud Computing*

[15] Linlin Wu, Saurabh Kumar Garg, Rajkumar Buyya, *SLA-based admission control for a Software-as-a-Service provider in Cloud computing environments*, Journal of Computer and System Sciences, 2012

[16] Wenhui Sun, Yue Xu, Feng Liu, *The Role of XML in Service Level Agreements Management*, Beijing, Network management Research Center