Alma Mater Studiorum · Università di Bologna

SCUOLA DI SCIENZE
Corso di Laurea in Informatica

Problematiche di trust e privacy negli ambienti di cloud computing

Relatore:
Chiar.mo Prof.
Fabio Panzieri

Presentata da:
Raffaele Pitaro

Sessione I
Anno Accademico 2014/2015

Introduzione

I numerosi investimenti nella modernizzazione della rete Internet e la diminuzione dei costi di manutenzione dei server, hanno caratterizzato, negli ultimi anni, un notevole incremento nell'utilizzo dei sistemi cloud. Con il termine "cloud computing" si intendono sia le applicazioni come servizi che sfruttano la rete, sia i sistemi hardware e software presenti nei data center che forniscono tali strumenti. Negli ultimi anni è ormai possibile utilizzare 1000 server per un'ora, grazie alla loro elasticità, scalabilità ed economicità, equivalentemente all'uso di un singolo server per 1000 ore. E' sempre più diffuso il concetto di "computing as a utility" [20]. In genere, tali servizi vengono denominati 'SaaS' (Software as a Service), 'IaaS' (Infrastructure as a Service) e 'PaaS' (Platform as a Service). I ricercatori, inoltre, della Hewlett-Packard (HP), a Bristol (UK), stanno portando avanti un prototipo di 'CaaS' (Cells as a Service), col quale sperano di automatizzare la gestione della sicurezza all'interno del cloud [21].

Questa tesi si prefigge l'obiettivo di analizzare alcuni aspetti critici della sicurezza in ambito cloud. In particolare, i problemi legati alla privacy, dai termini di utilizzo alla sicurezza dei dati personali più o meno sensibili. L'aumento esponenziale di dati memorizzati nei sistemi di cloud storage (es. Dropbox, Amazon S3) pone il problema della sensibilità dei dati su un piano tutt'altro che banale, dovuto anche a non ben chiare politiche di utilizzo dei dati, sia in termini di cessione degli stessi a società di terze parti, sia per quanto riguarda le responsabilità legali.

In diversi studi viene presentata una tassonomia dei problemi inerenti alla sicurezza nei cloud system. Tra i più preoccupanti possono essere annotati sicuramente quelli di sicurezza tradizionale, disponibilità dei servizi e controllo dei dati da par-

te di organismi di terze parti, tutti in qualche modo riconducibili a problemi già esistenti. A questi si aggiungono poi problemi nuovi che emergono dall'utilizzo dei sistemi cloud (es. analisi dei dati per produrre pubblicità). Bisogna considerare anche le diverse concezioni che si hanno riguardo alla privacy, a seconda della posizione geografica in cui si vive. Con la realizzazione di questo lavoro, attraverso un'analisi chiara e il più possibile dettagliata, si vogliono delineare gli strumenti di valutazione di privacy e trust nel cloud computing.

Considerando la privacy come un diritto fondamentale per ogni individuo e un necessario punto cardine per la costruzione e la modellazione dei sistemi cloud, questa tesi cerca di approfondire ed esaminare le mancanze più preoccupanti degli stessi. Oltre ad analizzare le principali preoccupazioni e i punti deboli dei servizi cloud, l'obiettivo di questo lavoro sarà quello di fare chiarezza sui passi e le infrastrutture che alcune aziende (es. Amazon) hanno implementato per avvicinarsi all'idea di 'safeness' nel cloud. Infine, l'ultimo obiettivo posto sarà l'individuazione di criteri per la valutazione/misura del grado di fiducia che l'utente può porre in questo contesto, distinguendo diversi criteri per classi di utenti (p.e, singolo utente finale, azienda, federazione di aziende).

Per l'elaborazione di questa tesi verranno presi in esame diversi articoli e studi scientifici che trattano le problematiche di trust e privacy nel cloud computing. Verrà quindi eseguita una revisione bibliografica, ampliando e approfondendo alcune tematiche presenti negli articoli, facendo riferimento ad alcuni casi e scenari reali citati negli stessi. Sarà attuato un confronto tra le diverse fonti, in modo da constatare le problematiche sotto più punti di vista, verificandone i punti concordanti e quelli discordanti, mediante una valutazione il più dettagliata possibile.

La tesi è strutturata in 4 capitoli: nel primo capitolo sarà presentata una tassonomia dei problemi presenti nei sistemi cloud, sia quelli classici degli ultimi decenni, anche se in una chiave moderna, sia problemi nuovi dovuti alla diffusione delle diverse architetture. Verranno presentati anche alcuni avvenimenti della storia recente, in cui queste problematiche sono affiorate in modo pesante, analizzandone cause ed effetti.

Nel secondo capitolo saranno trattate le strategie di 'safeness' adottate da alcu-

ne aziende, in ambito cloud. Inoltre, saranno presentate alcune possibili soluzioni, dal punto di vista architetturale, analizzando alcuni modelli studiati in diversi articoli: si vedrà come il ruolo dell'utente sarà di estrema importanza, mediante il controllo dei dati criptati o offuscati.

Il terzo capitolo sarà incentrato sulla ricerca di strumenti e metodi di valutazione che un utente, o gruppo di utenti, può utilizzare nei confronti di questi sistemi.

Infine, il quarto capitolo conterrà alcune considerazioni conlusive sul lavoro svolto e sui possibili sviluppi di questa tesi.

Indice

Capitolo 1

Analisi dei problemi nei cloud

In questo capitolo verranno presentate le principali problematiche di privacy e trust legate al cloud computing. In particolare, verranno analizzate sia questioni note in altri contesti e che si ripropongono nello sviluppo di questo nuovo modello, sia nuove criticità emerse con l'aumento dell' utilizzo dei servizi cloud. Infine verranno citati alcuni casi di cronaca recente, in cui alcune delle più grandi aziende attive nel settore sono state protagoniste.

1.1 Problemi classici

In una classificazione dei problemi relativi a privacy e trust nei sistemi cloud[1], ne vengono identificati alcuni, come ad esempio vulnerabilità delle macchine virtuali o dei servizi web, che sono essenzialmente vecchi problemi ripresentati in un nuovo contesto. Questa categoria di problematiche, in un certo senso, preoccupa meno di altre in quanto sono questioni già abbondantemente studiate e per le quali esistono le basi per delle soluzioni. In generale, per questi problemi, sono state individuate tre principali categorie di interesse che verranno analizzate qui di seguito.

1.1.1 Traditional Security

Questa categoria racchiude quelle problematiche come intrusioni e attacchi a computer o reti, le quali potrebbero essere facilitate dall'uso dei sistemi Cloud. In questa categoria sono incluse:

- **Attacchi a livello delle macchine virtuali**. Possibili vulnerabilità nelle macchine virtuali (VM) usate dai cloud provider, le quali potrebbero essere un potenziale problema per le architetture multi-tenant[1]. Un esempio, in questo caso, può essere quello di VMWare [2], mentre in [23] viene dimostrato come sia possibile mappare l'infrastruttura interna del cloud di Amazon, identificare dove si trova una particolare macchina virtuale e di conseguenza istanziarne di nuove in "co-abitazione" con quella individuata come obiettivo.

- **Vulnerabilità del cloud provider**. Sono le falle a livello piattaforma. Criticità di questo tipo sono state riscontrate in Google Docs [3].

- **Phishing cloud provider**. Il phishing è un tentativo di acquisire informazioni sensibili, ad esempio username, password e dettagli finanziari, spesso per cattive intenzioni. La maggior parte delle truffe avviene con messaggi da parte di "malintenzionati", che imitano aspetto e contenuto di quelli inviati dai fornitori di servizi in modo che gli utenti inseriscano i loro dati. La minaccia di phishing ha aumentato il suo potenziale d'attacco nel constesto dei Cloud system. Esempio, in tal senso, è l'incidente di Salesforce [4] che verrà descritto in seguito.

- **Expanded network attack surface**. Questa preoccupazione deriva dal fatto che gli utenti dovrebbero essere in grado di proteggere l'infrastruttura che gli permette di connettersi e interagire con il cloud. Quest'ultimo ha un ristretto campo d'azione in tal senso, poiché si trova "dall'altra parte" dei firewall.

[1] Multi-tenant si riferisce ad un'architettura software in cui una singola istanza software viene utilizzata da più di un "utente"

- **Autenticazione and Autorizzazione**. I framework utilizzati dalle aziende, in molti casi, non sono automaticamente estendibili ai sistemi Cloud. Inoltre, da questo punto scaturiscono un paio di riflessioni interessanti: come una compagnia modella i suoi framework esistenti per includere le risorse del cloud? Come fa un'azienda ad accordare le proprie politiche sulla sicurezza dei dati con quella dei cloud provider?

- **Forensics in the cloud**. Le tradizionali metodologie investigative, nel caso di crimini informatici, prevedevano il sequestro delle apparecchiature onde evitare la perdita di dati utili a causa di possibili sovrascritture. Ora, col cloud computing, un sequestro non garantisce il mantenimento al sicuro dei dati potenzialmente incriminati [5].

1.1.2 Availability

Questa categoria fa riferimento alla disponibilità di dati e applicazioni critiche. Casi di Cloud fuori servizio, come gli incidenti occorsi a Gmail nel 2008 [6] e Amazon S3 [7] saranno trattati maggiormente nello specifico più avanti. Le principali criticità che sono emerse in questo contesto sono:

- **Uptime**. I *Cloud provider* assicurano che l'uptime[2] dei loro server è migliore rispetto a quello di un data center privato. In questi casi le preoccupazioni derivano anche dai cloud di terze parti, i quali potrebbero non avere la potenza necessaria a gestire determinati picchi di carico o particolari applicazioni.

- **Single point of failure (SPOF)**. In questo caso, se una parte di un "meccanismo" fallisce, il funzionamento dell'intero sistema viene compromesso. In contesti dove sono richiesti *availability* e *reliability* questo rappresenta un problema non da poco. Tutto ciò viene maggiormente amplificato nel caso in cui ci siano più **SPOF** nello stesso sistema.

[2]Denota l'intervallo di tempo in cui un singolo apparato o un intero sistema informatico è stato ininterrottamente acceso e correttamente funzionante, contrario di downtime

- **Garanzia di integrità computazionale**. Non c'è piena garanzia, da parte di un Cloud provider, che un'applicazione lanciata da un'azienda su un sistema cloud venga eseguita correttamente e in modo accurato, consegnando di conseguenza dei risultati validi. In alcuni casi, si è pensato di ovviare a questa mancanza facendo eseguire lo stesso *task* a più client contemporaneamente, in modo da poter comparare poi i risultati ed avere un consenso più largo possibile sugli stessi.

1.1.3 Controllo dei dati da parte di terzi

Questo è considerato forse l'aspetto più ambiguo e complesso di questa categoria di problematiche, poiché l'aggiunta di implicazioni legali sulla gestione dei dati (più o meno sensibili) ha contribuito ad accrescere la pericolosità della questione. Tutto ciò implica una carenza di controllo e trasparenza sui dati quando essi sono gestiti o comunque sono in possesso di società terze. A causa di ciò, alcune compagnie stanno pensando di costruire dei cloud privati in modo da evitare questa problematica e mantenere i vantaggi dell'utilizzo del cloud computing. Gli aspetti principali in questo contesto sono:

- **Due diligence**. In caso di azioni legali da parte del *cloud user* nei confronti del *cloud provider* non è ben chiaro se e come il primo possa obbligare che il secondo risponda in un arco di tempo ragionevole. Allo stesso modo, nel caso in cui l'utente decidesse di cancellare i propri dati, nessuno gli assicura che le proprie informazioni saranno veramente cancellate o ancora trattenute dal cloud provider.

- **Verificabilità**. La mancanza di trasparenza e di controllo sulla gestione dei dati porta a dei problemi di verifica degli stessi. Perciò non si hanno molte informazioni sulle attività che il cloud provider esegue con i dati degli utenti. Inoltre, tra le richieste avanzate, in alcune alcune regolamentazioni c'è quella del mantenimento dei dati in determinate locazioni geografiche. In seguito verrà analizzato in modo più approfondito come i cloud provider rispondano a tale necessità [8].

- **Obblighi contrattuali**. Una norma del termini di utilizzo di Amazon rende molto bene questo concetto:

> 10.4. Non-Assertion. *During and after the term of the Agreement, with respect to any of the Services that you elect to use, you will not assert, nor will you authorize, assist, or encourage any third party to assert, against us or any of our customers, end users, vendors, business partners (including third party sellers on websites operated by or on behalf of us), licensors, sublicensees or transferees, any patent infringement or other intellectual property infringement claim with respect to such Services.*

  In poche parole, dopo che un utente utilizza il servizio EC2, non può esercitare nessun diritto di reclamo nei confronti di Amazon o di società collegate ad essa, anche nel caso in cui venissero violati non solo i dati personali, ma anche dei brevetti.

- **Spionaggio da parte del cloud provider**. Questo caso include la paura di furto di dati personali (sia di singoli utenti che di aziende) da parte del cloud provider. Nel caso di aziende, Google Gmail e Google Apps sono esempi di servizi basati su proprie infrastrutture cloud. Per quanto riguarda i singoli utenti, soprattuto all'inizio si è notata una forte preoccupazione per la confidenzialità dei propri dati (Gmail [9]). Col passare del tempo, però, sembra che gli utenti abbiano dissolto le loro preoccupazioni o, comunque, che il pericolo di salvare i propri dati nei cloud sia stato superato dall'utilità nell'uso del cloud stesso.

- **Data Lock-in**. Questo scenario si può riferire a casi in cui i dati siano bloccati in un formato proprietario, con conseguenze anche processuali, o a situazioni in cui le applicazioni di un cloud-user possano rimanere bloccate in seguito allo shutdown di una piattaforma cloud. In questo caso, spesso in mancanza di meccanismi standard, gli utenti sarebbero costretti a rimodellare le loro applicazioni per essere eseguite su un'altra piattaforma.

- **Transitività**. Spesso i cloud provider stringono accordi con altre compagnie, le quali potrebbero essere considerate non fidate dall'utente ma sulle quali egli non ha controllo. Casi come quello di Linkup e Nirvanix [13], descritto in seguito, in cui la cattiva gestione dei dati in un contesto nel quale la prima compagnia si appoggia ai servizi di storage della seconda, possono essere un perfetto esempio per evidenziare questa problematica.

1.2 Problemi emergenti

In questa sezione verranno analizzati alcuni problemi che potranno emergere o che sono solamente apparsi fino ad ora, ma che con la diffusione dei sistemi cloud saranno sicuramente oggetto di studio.

Con l'aumento dei cloud, sono stati creati degli enormi data-set per scopi di lucro, cioè vengono utilizzati per essere "monetizzati" e venduti a società terze per obiettivi pubblicitari. Ad esempio, Google utilizza le sue infrastrutture private per mantenere ed analizzare i dati degli utenti, in modo da poterli veicolare per fini pubblicitari. Tutto ciò è possibile poiché, in questo nuovo contesto, la "collezione" e analisi dei dati sono operazioni meno costose, anche per società che non siano Google. Inoltre poiché i dati, disponibili per essere estratti ed analizzati, vengono salvati in database, questi posso facilmente essere preda di attacchi.

In seguito a sempre più diffuse richieste di privacy, le aziende proprietarie hanno subito pressioni per anonimizzare i dati in loro possesso. Inoltre, per quanto riguarda i "dati di ricerca" degli utenti, devono essere resi anonimi dopo 18 mesi dalla loro memorizzazione. Questo implica che alcune informazioni identificative (come indirizzo ip e cookie) verranno rimosse, mentre altri dati verranno mantenuti in forma anonima per testare gli algoritmi di sicurezza.

Lo scenario di *availability* esaminato in precedenza deve essere considerato anche in caso di attività di sabotaggio. I danni in tal senso non sono solo relativi ad una perdita di produttività, bensì anche alla diminuzione di fiducia nei con-

fronti dell'intera infrastruttura e ad un aumento dei costi per opportune misure di backup. Poiché sovente, nei sistemi cloud, vengono riscontrati problemi dovuti a SPOF (Single Point Of Failure), è importante sviluppare metodi per un'efficiente availability e per sistemi di recovery in caso di attacchi esterni.

Lo spostamento di sempre più servizi nei cloud sta portando ad un alleggerimento delle "responsabilità" da parte degli end-user, poiché essi delegano molti compiti ai provider. Piuttosto che installare un software, gli utenti preferiscono utilizzare delle applicazioni direttamente nei cloud. Gli effetti di questo scenario sono diversi e possono spaziare da una monitorizzazione dei rischi più centralizzata, e teoricamente più efficiente, alla possibilità di evitare, almeno in parte, la diffusione di dati sensibili su client non fidati. Tuttavia, questo modello aumenta la necessità di sistemi di autenticazione sicuri. Inoltre, la crescente mole di dati e applicazioni mantenute nei cloud e la minor dipendenza dalle macchine fisiche degli utenti, comporta una maggior esposizione a minacce che hanno come obiettivo l'acquisizione di credenziali d'accesso, come phishing o metodi di crittoanalisi come gli attacchi brute-force[3].

Man mano che aumenta l'uso del cloud computing, si nota un incremento anche di servizi che effettuano *mash-up* dei dati. Le potenziali implicazioni di questo scenario possono essere notevoli, sia in termini di perdita di dati che per quanto riguarda le fonti dalle quali un utente può estrarre dati. Questo scenario pone delle questioni su come viene autorizzato l'accesso per ragioni di usabilità. Un esempio in tal caso viene dato da Facebook. I suoi utenti aggiornano dati più o meno sensibili, i quali vengono utilizzati come "presentazione" per altri utenti, ma anche da applicazioni terze le quali, spesso, non sono verificate da Facebook.

[3]Consiste nel testare sistematicamente tutte le possibili chiavi o password di un qualsiasi sistema, finché non viene trovata quella corretta. In caso di password brevi e semplici è un metodo abbastanza veloce, tuttavia per password più lunghe e complicate risulta lento e dispendioso, perciò è utilizzato come ultima opzione e gli vengono preferiti altri tipi di attacchi.

1.3 Casi di storia recente

Negli ultimi anni, diverse criticità nei cloud system delle grandi aziende sono venute alla luce. Spesso, in seguito ad improvvisi shutdown dei sistemi, dati e informazioni degli utenti sono diventati inaccessibili e, in alcuni casi, perfino perduti. I problemi sono stati riscontrati in diverse aree tra quelle citate sopra.

Ad esempio, nel caso delle macchine virtuali, un utente di VMWare, nel 2008 [2], è riuscito ad effettuare un accesso remoto ottenendo privilegi elevati all'interno del sistema. In particolare, vCloud Automation Center (vCAC) aveva un'escalation di privilegi da remoto e l'utente, sfruttando una vulnerabilità nella VMware Remote Console (VMRC) è riuscito ad ottenere l'accesso come amministratore al vCenter Server. In seguito è stato riscontrato che il bug era presente utilizzando la funzione "Connect (by) Using VMRC" per connettersi direttamente al vCenter Server, mentre non sono stati rinvenuti casi critici in cui veniva usato vCloud Director come proxy per connettersi. Questa, tuttavia, non è una garanzia del fatto che la connessione con vCloud Director sia più sicura, ma semplicemente che, in questo scenario, non sono stati rilevati problemi mediante l'utilizzo del proxy e quindi non è stato considerato nello scope del problema. In seguito l'azienda ha rimosso la possibilità di connessione mediante VMRC, lasciando attive quelle che facevano uso di RDP e SSH. Casi analoghi sono stati scoperti su Xen [10] e VirtualPC di Microsoft [11].

A livello di piattaforme, degno di nota è un bug riscontrato in Google Docs [3], il quale ha permesso che dei documenti privati venissero divulgati. L'incidente, che secondo l'azienda era dovuto ad una sequenza di operazioni effettuate dai proprietari dei documenti, ha coinvolto solo una piccola percentuale di utenti e ha avuto come conseguenza una condivisione incontrollata del loro materiale.

Sempre per la stessa azienda, è doveroso citare l'interruzione per 24 ore dei servizi di Gmail [6], che ha impedito ai propri utenti di accedere ai sistemi. Anche in questo caso, la compagnia ha affermato che la percentuale di vittime di questo incidente era molto bassa, non fornendo per diverso tempo alcuna spiegazione sul motivo del malfunzionamento.

Nei casi di phishing può essere annoverato quello relativo a Salesforce.com [4], nel 2007. A numerosi utenti sono state inviate email false in seuito all'acquisizione illecita di dati rubati all'azienda. L'incidente ha avuto inizio quando un impiegato dell'azienda, vittima anch'esso di una falsa email, ha usato una password aziendale, consegnando così l'accesso ad una lista di contatti e altre informazioni degli utenti. In seguito, la raccomandazione di Salesforce.com ai propri utenti è stata quella di rafforzare le misure di sicurezza, restringendo l'accesso agli account, effettuando una preparazione al problema del phishing e utilizzando delle tecniche di autenticazione più sicure per connettersi ai server.

Amazon S3 è un esempio del problema di availability [7]. Dal momento in cui si è notato che il grado di errore nei datacenter era elevato e che poche richieste venivano completate con successo, si è provato ad abbassare il carico di processi ma non si è riusciti a ristabilire le normali condizioni del sistema. In seguito si è capito che i server di Amazon S3 stavano avendo problemi a comunicare con tutti gli altri server. Utilizzando un protocollo di "investigazione[4]", si è constatato che un gran numero di server stava spendendo quasi tutte le sue risorse investigando, mentre molti più server avevano fallito nell'eseguire la stessa mansione. Decretata l'impossibilità ad eseguire la maggior parte delle richieste, Amazon ha deciso di "spegnere" tutte le comunicazioni tra i server, disattivare tutti i componenti utilizzati per processare le richieste e cancellare lo stato del sistema. Il motivo del malfunzionamento pare sia stata trovato in alcuni messaggi che, avendo un singolo bit corrotto che non influiva sulla leggibitità degli stessi, abbiano alterato le informazioni sullo stato del sistema causando così problemi di comunicazione server-to-server.

[4]Amazon utilizza un protocollo per verificare lo stato dei server attraverso il sistema. Quando un server si connette ad un'altro server per processare una parte di richiesta utente, esso inizia ad investigare sullo stato del sistema e, completate le "investigazioni", tutte i risultati vengono spediti al server "richiedente".

Coghead, un servizio web-based per la costruzione e l'hosting di database applicativi, annunciando il proprio fallimento [12] ha sollevato il grosso problema che i dati e le applicazioni degli utenti rimanessero bloccati e non fossero più estraibili. L'effetto di tutto ciò è stato il precipitarsi degli utenti per trovare delle valide alternative a Coghead ed esportare i propri dati dalle varie applicazioni. Gli utenti hanno avuto la possibilità di estrarre i propri dati, ma le applicazioni costruite e modellate su Coghead non sono state più utilizzabili, costringendoli così a dover riscrivere i loro software in modo da poter funzionare in altri contesti.

Infine, un caso in cui più della metà dei dati degli utenti sono andati definitivamente persi, e che rientra nello scenario di "Transitive nature", è quello di Linkup (MediaMax precedentemente), nel 2007 [13]. Poiché era prevista una migrazione di alcune applicazioni e database, mantenuti da Linkup, verso i sistemi di Nirvanix, è molto probabile che queste operazioni abbiano condotto ad una cattiva gestione dei dati e alla perdita di essi. Dopo un lungo rimbalzo di colpe tra le due compagnie, su chi delle due fosse in possesso dei dati e avesse potuto recuperarli, la questione non è stata ancora del tutto chiarita.

Capitolo 2

Politiche aziendali e nuovi modelli

In questo capitolo si cercherà di analizzare le politiche che le grandi compagnie hanno adottato per aumentare il livello di privacy e trust nei loro sistemi. In particolare, verranno evidenziate le misure che sono state prese da alcune aziende per arginare o porre rimedio alle falle di sicurezza riscontrate nel tempo. Inoltre, considerato l'ancora ampio margine di manovra nella direzione di sistemi più sicuri e fidati, saranno presentati alcuni possibili scenari che potrebbero portare a risultati non da poco nella risoluzione di queste problematiche.

2.1 IBM Rational AppScan tool

Ad inizio 2009, IBM ha rivelato una serie di nuovi prodotti, servizi, client e partnership per la sua iniziativa "Blue Cloud", attraverso la quale cerca di sviluppare soluzioni cloud integrate [14]. L'IBM Rational AppScan aiuta le compagnie ad assicurare che i servizi Web che vengono sviluppati all'interno del cloud siano sicuri. In particolare, questa tecnologia fornisce strumenti di valutazione sulla sicurezza delle applicazioni Web, esegue la scansione di vulnerabilità comuni delle applicazioni, genera dei report processabili e aiuta a gestire le norme e gli standard nell'ambiente "online".

Utilizzando la suddetta applicazione, gli utenti (intesi anche come aziende), possono mitigare e abbassare i costi di gestione dei propri servizi Web. In que-

sto caso IBM è considerata l'unica azienda che offre soluzioni di sicurezza che si estendono su tutte le aree di distribuzione delle applicazioni [15]:

- **Fornire sicurezza per le applicazioni sviluppate nel cloud**. Le compagnie possono ridurre i rischi, assicurandosi che i servizi Web che vengono "pubblicati" nel cloud siano sicuri, conformi e vadano incontro alle politiche di business. Inoltre, utilizzando IBM Rational AppScan On Demand, le aziende possono anche assicurare che i servizi Web vengano monitorati continuamente, fornendo delle analisi di sicurezza ai gestori delle applicazioni.

- **Mettere le aziende al riparo dalle vulnerabilità derivanti da Web 2.0 e SOA[1] Vulnerabilities**. Abilita le compagnie a scansionare e testare i contenuti flash e controlla che non ci siano problemi di sicurezza, prima che questi possano diventare una minaccia reale.

- **Aiutare le organizzazione ad adottare un approccio preventivo alla sicurezza**. Permette alle organizzazioni di testare le vulnerabilità dal momento in cui l'applicazione viene progettata fino a dopo esser stata sviluppata, cioè prima che possa rappresentare un rischio reale per gli utenti e la compagnia e che la riparazione diventi altamente costosa.

- **Gestire il continuo rischio di conformità e sicurezza**. Attraverso capability di valutazione dei rischi, IBM permette agli utenti di comprendere meglio dove sono localizzate le vulnerabilità di sicurezza, e di conseguenza eseguire una qualche azione che elimini la possibile falla e probabili rischi futuri. Inoltre, con l'ausilio dei meccanismi di monitoraggio dell'AppScan On Demand, essi possono anche catturare e essere avvisati di vulnerabilità esistenti, rendendo più semplice e veloce la riparazione delle falle e rimanere in regola.

[1] Service-oriented Architecture

2.2 Amazon

Amazon Web Services (AWS) offre una piattaforma scalabile di cloud computing con alta disponibilità, affidabilità e flessibilità in modo da permettere agli utenti di costruire un'ampia gamma di applicazioni. I processi di sicurezza fisici ed esecutivi sono descritti per le infrastrutture di rete e i server sotto la gestione di AWS, in aggiunta alle implementazioni di sicurezza per i servizi specifici. L'utilizzo delle infrastrutture di AWS crea un modello di responsabilità condivisa tra l'utente e il provider. Questo modello può ridurre il carico di lavoro dell'utente poiché AWS opera, gestisce e controlla i componenti dal sistema operativo dell'host e il livello di virtualizzazione fino alla sicurezza fisica delle strutture in cui si trova il servizio. L'utente si assume la responsabilità e la gestione del sistema operativo guest (inclusi gli aggiornamenti di sicurezza), altre appplicazioni software associate e la configurazione del firewall fornito da AWS. L'utente può migliorare il livello di sicurezza e/o attuare misure più stringenti facendo leva su tecnologie come firewall lato host, tecniche di prevenzione/rilevamento minacce e sistemi di crittografia. AWS organizza l'infrastruttura cloud in modo da fornire una varietà di risorse di computazione di base come l'elaborazione di dati e lo storage.

I data center di Amazon sono dislocati in varie regioni geografiche. In caso di malfunzionamenti, alcune procedure automatizzate spostano il traffico dati degli utenti dalla zona interessata. L'azienda permette di localizzare le istanze e salvare i dati in diverse aree geografiche o in più zone, ognuna strutturata come indipendente, nella stessa area, in modo così da permettere agli utenti di distribuire le proprie applicazioni.

La rete di AWS è stata architettata per permettere all'utente di scegliere il livello di sicurezza ed elasticità appropriato per il suo lavoro. Le periferiche di rete (p.e. firewall) vengono usate per monitorare e gestire le comunicazioni verso l'esterno. Per gestire e rafforzare il flusso di traffico, vengono utilizzate le ACL (access control list) o altre politiche e configurazioni da settare su ogni interfaccia gestita.

AWS utilizza una vasta gamma di sistemi di monitoraggio automatizzati. Questi strumenti permettono di controllare l'uso dei server e della rete, effettuano la

scansione delle attività delle porte e verificano l'utilizzo delle applicazioni. Inoltre, permettono di settare delle metriche appropriate per le attività inusuali e per i tentatvi di intrusione non autorizzati. In particolare, forniscono una protezione significativa contro le tradizionali minacce di sicurezza in rete, come attacchi DDos (Distributed Denial Of Service) o MITM (Man in the Middle Attacks), IP spoofing, port scanning e packet sniffing, permettendo all'utente di implementare ulteriori sistemi di protezione. Anche il sistema operativo host, le applicazioni web e i database vengono regolarmente monitorati e scansionati per cercare possibili vulnerabilità. Gli sviluppatori e amministratori AWS che hanno necessità di accedere ai componenti cloud, devono esplicitamente fare richiesta attraverso il sistema di gestione degli accessi. Tutte le richieste devono essere visionate e approvate dal proprietario o gestore di riferimento. Gli account vengono revisionati ogni 90 giorni, al termine dei quali è richiesta un'altra approvazione, altrimenti l'accesso alle risorse viene automaticamente revocato.

Prima di descrivere più dettagliatamente le pratiche di sicurezza attuate nei servizi di Amazon, è doveroso citare AWS IAM (AWS Identity and Access Management), il quale permette all'utente di creare diversi profili e gestire i permessi, per ognuno di essi, all'interno del suo account AWS. Un profilo è un'identità con delle credenziali di sicurezza uniche utilizzabili per accedere ai servizi. Questo sistema elimina la necessità di condividere password o chiavi, rendendo più semplice l'abilitazione o disabilitazione all'accesso. Inoltre, abilita l'utente ad implementare migliori pratiche di sicurezza, come ad esempio l'utilizzo dei privilegi, concedendo credenziali uniche ad ogni utente all'interno dello stesso account AWS e i permessi di accedere ai servizi e alle risorse richieste per permettere agli stessi di eseguire il proprio lavoro. IAM è integrato con il Marketplace AWS, così l'utente può controllare chi, nella sua organizzazione, aderisce ai software e ai servizi offerti nel Marketplace.

2.2.1 Amazon EC2

Amazon Elastic Compute Cloud (EC2) è un componente chiave in Infrastructure as a Service (IaaS) di Amazon, che fornisce capacità computazionale utilizzando

istanze server nei data center di AWS. Le istanze sono collezioni di piattaforme hardware e software. La sicurezza all'interno di EC2 è composta da diversi livelli, illustrati in figura 2.1: il sistema operativo (OS) della piattaforma host, l'istanza virtuale dell'OS o quello di un guest, un firewall e chiamate ad alcune API[2]. Ogni livello fa "affidamento" su quello sottostante. L'obiettivo di tale struttura è quello di prevenire che i dati contenuti siano, in qualche modo, intercettati da sistemi o utenti non autorizzati, fornendo inoltre delle istanze il più possibile sicure senza effetturare sacrifici in termini di flessibilità delle configurazioni.

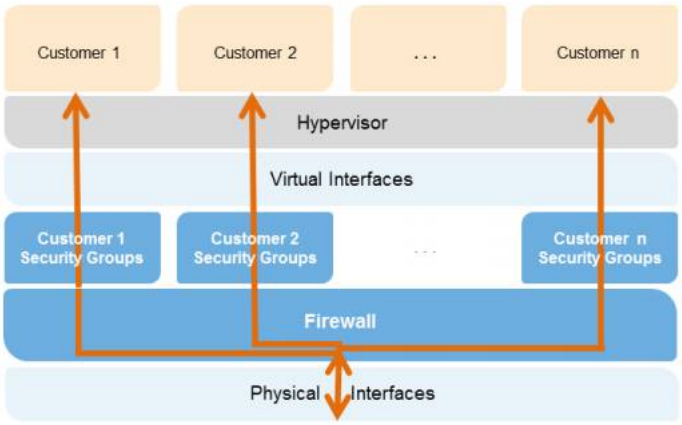

Figura 2.1: Struttura a livelli di Amazon

Amazon EC2 utilizza una versione personalizzata dell'*hypervisor* di Xen, sfruttando la paravirtualizzazione nel caso di ospiti Linux. Questa è una tecnica di virtualizzazione che presenta un'interfaccia software alle macchine virtuali simile - ma non uguale a - quella dell'hardware sottostante. Il motivo di un'interfaccia modificata è da ricercare nella necessità di ridurre il tempo di esecuzione delle operazioni di un guest, le quali sono più difficili da eseguire in uno scenario virtualizzato rispetto ad uno non virtualizzato. Questa tecnica fornisce delle particolarità chiamate "hooks" per permettere ai guest e agli host di richiedere e riconoscere questi processi, che altrimenti potrebbero essere eseguiti in un dominio virtuale, dove cioè

[2]Le Application Programming Interface sono una serie di routine, protocolli e tool per la costruzione di applicazioni software. Un' API esprime un componente software nei termini delle sue operazioni (input, output etc.).

le performance di esecuzione sono peggiori. Il sistema operativo non ha un elevato accesso alla CPU, poiché un guest paravirtualizzato fa affidamento sull'hypervisor per fornire supporto alle applicazioni che normalmente richiedono dei privilegi di accesso. La CPU fornisce quattro diversi livelli, chiamati *ring* di privilegi che vanno da 0 a 3, con 0 il più privilegiato. Il sistema operativo host ha privilegi 0, tuttavia piuttosto che eseguire a questo livello, come fanno molti sistemi operativi, L'OS guest lavora a livello 1 e le applicazioni al 3. Questa virtualizzazione delle risorse fisiche porta ad una netta separazione tra guest e hypervisor, avendo come conseguenza un'ulteriore separazione di sicurezza tra i due.

Ogni istanza eseguita sulla stessa macchina virtuale viene isolata dalle altre con l'ausilio dell'hypervisor Xen. Il firewall di AWS risiede all'interno del livello dell'hypervisor e, poiché tutti i pacchetti devono passare attraverso esso, le istanze vicine non hanno maggiore accesso a quella istanza rispetto ad un qualsiasi host sulla rete e quindi possono essere trattate come se fossero su degli host fisici separati. La RAM fisica viene separata utilizzando un meccanismo simile. Le istanze dei clienti non hanno accesso ai device primari, i quali invece sono presentati come dischi virtualizzati. Il layer di virtualizzazione dischi resetta automaticamente ogni blocco di storage usato dal cliente, così che una sua informazione non sia mai involontariamente visibile ad altri. Inoltre, la memoria allocata per i guest, nel caso in cui sia deallocata, viene cancellata (settata a 0) dall'hypervisor e non ritorna ad essere disponibile per nuove allocazioni fino al termine del processo di cancellazione. AWS raccomanda ai propri clienti una migliore protezione dei loro dati utilizzando degli opportuni strumenti come, ad esempio, avere un file system cifrato sopra il disco virtualizzato.

Gli host OS sono sistemi specificatamente designati, costruiti e configurati per proteggere l'area manageriale del cloud. Tutti gli accessi sono registrati e verificati. Quando un impiegato non ha più bisogno di accedere all'area di gestione gli vengono revocati i privilegi di accesso a questi host. Mentre i guest OS sono delle istanze completamente controllate dal cliente, il quale ha pieno controllo amministrativo sugli account, servizi e applicazioni. AWS raccomanda un insieme di consigli di sicurezza di base che includono la disabilitazione dell'accesso con sola password

per gli ospiti e l'utilizzo di una qualche forma di autenticazione a più fattori per autenticarsi con le istanze (o almeno SSH versione 2 con accesso basato su certificati). Inoltre, il cliente può impiegare un'escalation di privilegi. Per esempio, se l'OS guest è Linux, si può utilizzare un accesso alle istanze virtuali basato su SSHv2, disabilitare il login root da remoto, e usare sudo per avere un'escalation di privilegi. Il cliente può anche generare una propria coppia di chiavi in modo da garantire l'unicità delle stesse e la non condivisione con altri clienti o con AWS.

Il firewall di Amazon EC2, non controllato attraverso il guest OS, è impostato di default in modalità deny-all così che i clienti possano esplicitare direttamente le loro impostazioni per il traffico, il quale può essere può essere ristretto a seconda del protocollo, delle porte o edgli indirizzi IP sorgenti. AWS supporta la possibilità di assegnare un accesso "granulare" a differenti funzioni amministrative su istanze e firewall, quindi abilita il cliente ad implementare un livello addizionale di sicurezza attraverso la separazione dei compiti.

2.2.2 Amazon Simple Service Storage

L'azienda, sempre nel 2008, ha fornito un documento in cui esegue una panoramica delle sue politiche di sicurezza in generale [16]. Per quanto riguarda il servizio Amazon Simple Service Storage (S3), che permette di salvare dati come oggetti (file di testo, foto, video ecc.) all'interno di un *bucket*, all'utente che aggiunge un file viene data la possibilità di includere i metadati con il file stesso, e settare i permessi per controllare gli accessi al file. Per ogni bucket, l'utente può controllarne gli accessi (chi può creare, cancellare o elencare oggetti), verificarne i log di accesso ai suoi oggetti e sceglierne la locazione geografica.

L'accesso ai dati è ristretto per default, solo i proprietari di bucket e oggetti possono accedere alle risorse di Amazon S3 create da loro (il proprietario di un bucket/oggetto è lo stesso dell'account AWS[3] e non l'utente che lo ha creato). Per controllare chi accede ai bucket si può fare uso di:

[3]Amazon Web Service.

- **Politiche di identificazione e gestione degli accessi (IAM)**. Permette alle organizzazione con molti impiegati di creare e gestire diversi account utenti sotto un singolo account AWS. Questa politica consente un controllo centralizzato dei permessi e, a discrezione del proprietario dell'account AWS, abilita l'accesso agli utenti.

- **Lista di controllo accessi (ACL)**. Le ACL, in genere utilizzate coi file system, sono liste di permessi allegate ad un oggetto. Una *access control list* specifica quali utenti o processi di un sistema possono accedere ad un oggetto, e quali operazioni possono essere compiute su di esso. In Amazon S3 vengono utilizzate per dare permessi di lettura o scrittura su un bucket ad un utente o un gruppo di utenti, con la restrizione di poter invitare solamente altri account AWS e non specifici utenti.

- **Politiche sul bucket**. Vengono utilizzate per aggiungere o negare permessi su alcuni o tutti gli oggetti dentro un singolo bucket. Queste politiche possono essere applicate ad utenti, gruppi o ai bucket stessi, eseguendo così una gestione centralizzata dei permessi. A differenza degli altri casi, questo meccanismo può essere applicato sia ad altri account AWS che agli utenti "dentro" il proprio.

Inoltre, sotto determinate condizioni si può limitare l'accesso a delle specifiche risorse. Ad esempio, è possibile effettuare restrizioni in base al tempo richiesto (Date condition), l'utilizzo o meno di SSL (Booloean conditon), l'indirizzo IP utilizzato dal richiedente (IP Address Condition) oppure a seconda dell'applicazione client utilizzata (String condition).

Amazon fornisce diverse opzioni per la protezione dei dati. Per gli utenti che vogliono gestire le loro chiavi di cifratura è possibile utilizzare una libreria client come *Amazon S3 Encryption Client* per criptare i dati prima di fare l'upload su Amazon S3. In alternativa, se si preferisce delegare la criptatura al service provider si può utilizzare *Amazon S3 Server Side Encryption* (SSE), la quale cripta i dati in upload semplicemente aggiungendo un header "request" al momento di scrivere

l'oggetto. La decriptazione avviene automaticamente quando si recuperano i dati. In entrambi i casi i metadati non vengono criptati. Con Amazon S3 SSE, ogni oggetto protetto è cifrato con un'unica chiave di criptazione, la quale poi viene a sua volta criptata con una "master key" regolarmente cambiata. La chiave e i dati criptati vengono salvati in differenti host. È anche possibile creare delle politiche da applicare al bucket in modo che sia concesso eseguire upload solo di dati criptati.

2.2.3 Post incidente 2008

In seguito agli avvenimenti del 2008, analizzati precedentemente, è emerso che Amazon effettua un checksum MD5[4] in ogni parte del sistema [7], ad esempio per prevenire, rilevare ed effettuare il *recovery* del sistema dalla corruzione che può occorrere durante la ricezione, lo storage e il recupero dei dati degli utenti. Tuttavia, il punto debole della vicenda è stato non avere la stessa protezione per rilevare se le informazioni sullo stato interno del sistema fossero corrotte. Di conseguenza, il problema non è stato rivelato prontamente e il danno, come descritto nel capitolo precedente, si è sparso velocemente per tutto il sistema.

Poiché è stato il primo caso in cui la compagnia si è dovuta imbattere in una criticità che coinvolgesse la comunicazione server-to-server in modo così massiccio, è stato necessario prendere un po' di tempo, sia durante la crisi per diagnosticare e risolvere il problema, sia successivamente all'incidente in modo da poter prevenire eventuali situazioni analoghe. Perciò le azioni intraprese sono state:

- Effettuare diversi cambiamenti in Amazon S3 che potessero ridurre significativamente il tempo richiesto per recuperare completamente lo stato dell'intero sistema e far ripartire la computazione delle richieste degli utenti.

[4]Il *message-digest algorithm* (MD5) è una funzione di crittografia hash che produce un output (checksum) di 128 bit, comunemente utilizzato per verificare l'integrità dei dati. La codifica è largamente usata poiché impiega poche risorse ed è altamente improbabile ottenere con due diverse stringhe in input uno stesso valore hash in output.

- Sviluppare una diversa tecnica di investigazione di Amazon S3 nel caso di guasti sui server, così da ridurre la quantità di investigazioni e facilitare la prevenzione di incidenti di questo genere.

- Aggiungere un'ulteriore attività di monitoraggio e notifica per quanto riguarda le soglie di investigazione e fallimento.

- Aggiungere il checksum per rilevare dinamicamente la corruzione dei messaggi di stato del sistema, in modo da poter accedere a tali messaggi, analizzarli e poi scartarli.

2.3 Soluzioni proposte

In diversi studi vengono avanzate delle proposte di possibili soluzioni per migliorare le condizioni di privacy e trust nei cloud system. Riprendendo il discorso trattato nel capitolo precedente, in un cloud pubblico, il fatto che gli utenti possano delegare l'amministrazione di un sistema al cloud provider vuol dire anche che la qualità dell'amministrazione o delle operazioni non può essere controllata dagli utenti. Inoltre, in un servizio multi-tenant, grazie alle tecnologie di virtualizzazione si presentano degli ulteriori problemi su dove i dati vengano "fisicamente" allocati e sulla protezione contro possibili minacce sullo stesso scenario virtuale. Se si guarda all'insicurezza del cloud dal punto di vista di quella sociale, poiché uno dei maggiori impedimenti nell'adozione di tecnologie cloud è proprio la paura di carenze di sicurezza all'interno di questi sistemi, in [25] vengono proposti alcuni concetti per costruire cloud sicuri. In [17], presentando un modello di cloud "trust", l'insicurezza sociale viene divisa in: la presenza di diversi stakeholder[5], la sicurezza "dell'open space" e il problema della gestione dei dati. Analizzandoli più dettagliatamente:

[5]Tutti i soggetti, individui od organizzazioni, attivamente coinvolti in un'iniziativa economica (progetto, azienda), il cui interesse è negativamente o positivamente influenzato dal risultato dell'esecuzione, o dall'andamento, dell'iniziativa e la cui azione o reazione a sua volta influenza le fasi o il completamento di un progetto o il destino di un'organizzazione.

- **Molteplici stakeholder**. Di solito, in una organizzazione, i dati vengono gestiti sotto il controllo dell'organizzazione. Al contrario, in un cloud non c'è un singolo agente di controllo interno, bensì ci sono almeno tre "agenti" da considerare: gli impiegati di una determinata organizzazione, il cloud service provider e le terze parti, ovvero gli stakeholder e possibili portatori di minacce. In questo caso l'azienda delega automaticamente parte dei suoi sistemi di amministrazione al cloud provider e, nonostante essa voglia controllare le operazioni delegate con i suoi stessi criteri, il provider agisce sotto politiche differenti rispetto al cliente. Poiché il contratto gioca un ruolo di collante tra le due parti, in un tale contesto piuttosto che rinforzare alcuni alcuni aspetti di sicurezza, ne mina la credibilità e causa insicurezza sociale. Inoltre, in un cloud pubblico, i dati possono essere visitati da chiunque sia autenticato, poiché la fase di autenticazione non è più sotto il controllo dell'aziende, ma del cloud provider.

- **Sicurezza dell'open space**. In un contesto generale, il controllo dei dati è implementato nel sistema dove essi sono localizzati ed avviene tenendo conto di due aspetti: dove il sisitema risiede e il modo in cui è amministrato. Invece, in un cloud dove spesso gli utenti non possono specificare dove localizzare i dati, il controllo degli stessi deve essere implementato in maniera differente. La via più sicura sarebbe attuare delle politiche di criptazione dei dati all'interno del cloud, ponendo l'attenzione sulle modalità di gestione delle chiavi.

- **Gestioni dei dati sensibili**. Anche in questo caso, a causa delle insicurezze nei confronti dei cloud, spesso i dati di importanza critica non vengono gestiti in un cloud. Una possibile soluzione potrebbe essere l'utilizzo di cloud privati, costruiti all'interno delle aziende proprietarie dei dati. In questo caso l'unico stakeholder sarebbe la compagnia stessa, e potrebbe attuare delle politiche di sicurezza e di controllo interno complete e coerenti. Tuttavia, la costruzione di un cloud privato è un'operazione costosa per gran parte delle aziende.

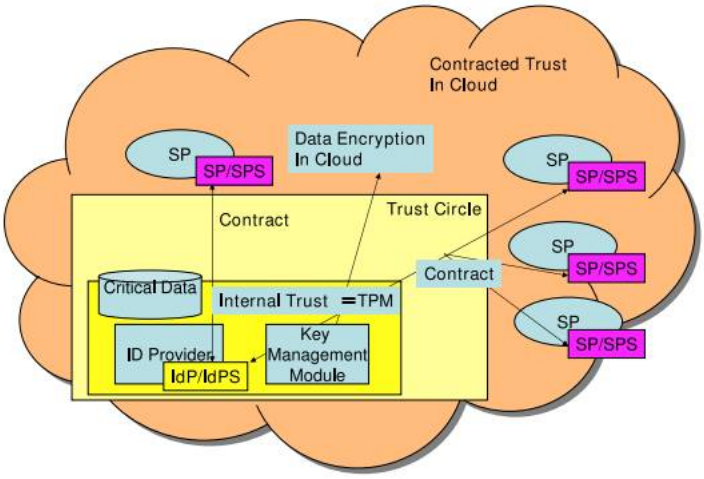

Figura 2.2: Modello di cloud trust

Nell'articolo si propone di risolvere questi problemi costruendo un modello gerarchico di trust nel caso di molteplici stakeholder. Considerando lo scenario tradizionale secondo il quale un'entità può essere all'interno o all'esterno di una "cerchia" trust, nel tentativo di costruire un modello cloud di fiducia vengono aggiunti due livelli di sicurezza che vorrebbero migliorare le politiche di controllo degli accessi *internal trust* e *contracted trust* layer (figura 2.2). Il primo è inteso come una piattaforma in cui ci sia garanzia che l'amministrazione e le operazioni effettuate al suo interno siano sotto il controllo di un'organizzazione, mentre il secondo tratta il significato di fiducia come derivante dal contratto vero e proprio. Se l'azienda deve controllare alcune operazioni, esse devono essere eseguite in internal trust. In questo scenario, si assume che la gestione degli Id, che autenticano un utente, così come quella delle chiavi per la criptazione siano nell'internal trust dell'azienda e quindi gestiti da essa.

Allo stesso modo, un cliente vuole controllare un cloud provider che opera sotto politiche diverse dalle sue e, quindi, si trova in una posizione intermedia tra il trusted e l'untrusted. L'accesso ad una cerchia trust può essere regolato dal contracted trust, che consiste essenzialmente di tre documenti:

- **Service Policy/Service Practice Statement (SP/SPS)**. Definisce la qualità del servizio dato dal provider. Un'organizzazione può entrare in una cerchia trust se accetta le condizioni di questo contratto e può fidarsi della sicurezza di un servizio per le entità che indicano SP/SPS.
- **Id Policy/Id Practice Statement (IdP/IdPS)**. Definisce la qualità dell'Id provider dato da un'organizzazione. Un cloud può fidarsi di un Id provider per le entità che indicano IdP/IdPS.
- **Contract**. Dichiara che un'organizzazione (O) usa i servizi di un cloud service provider (C) evidenziando che O ha accettato i termini di SP/SPS e C concorda su quelli di IdP/IdPS.

Nella negoziazione di un contratto, un'organizzazione può voler richiedere dei servizi con un livello di sicurezza più basso, a seconda dell'importanza dei dati, con una conseguente diminuzione dei costi, e un cloud service potrebbe richiedere un pagamento extra per dei servizi di sicurezza più elevati.

Una piattaforma cloud così costituita viene denominata "Security Aware Cloud" (figura 2.3). In tale modello sarebbero risolti i problemi degli Id e della gestione delle chiavi, poiché verrebbero gestiti in internal trust, e anche quelli derivanti dalla presenza di molteplici stakeholder e dell'open space.

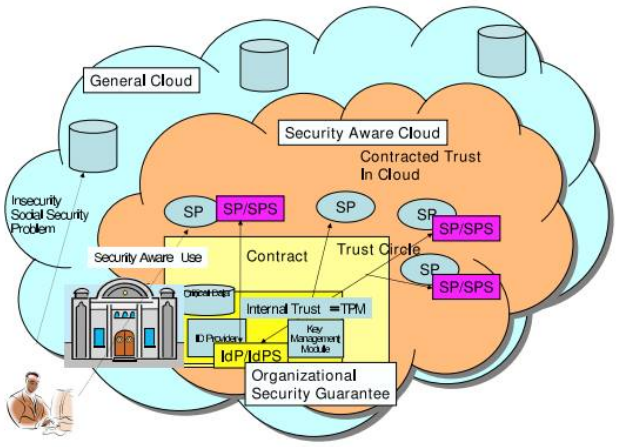

Figura 2.3: Security Aware Cloud

In un altro articolo [18] in cui vengono analizzate alcune problematiche legate al cloud computing, si evidenzia maggiormente il pericolo di perdita di dati e privacy in uno scenario non legato ad aziende, ma più incentrato sui singoli utenti. Spesso, i dati degli utenti vengono raccolti in modo da fornire loro dei servizi il più possibile personalizzati. Ad esempio, un utente può ricevere una notifica per sapere se qualche suo amico si trova nelle vicinanze, con conseguente inoltro di dati personali. In uno scenario simile le possibili minacce possono essere:

- Le informazioni personali di un utente possono essere raccolte, usate, salvate e/o propagate in modi che sono contro la volontà dell'utente stesso;
- Le Persone potrebbero avere accesso non autorizzato ai dati sensibili nel cloud, approfittanto di vulnerabilità come carenza di controlli di accesso, dati esposti in chiaro, politiche modificabili da entità non autorizzate e copie non protette di dati sparsi per il cloud;
- I flussi di dati che varcano i confini di qualche regione geografica potrebbero essere soggetti a regolamentazioni aggiuntive, e quindi portare a un mancato rispetto della legge. Per esempio le legislazioni sulla privacy in India e nell'Asia in generale sono diverse molto diverse da quelle europee [19].

Inoltre, dalla legislazione in materia di privacy emergono una serie di richieste: una riduzione di dati sensibili e personali usati e salvati all'interno di un'infrastruttura cloud, una protezione dei dati stessi, una limitazione agli scopi per i quale i dati vengono utilizzati (devono essere utilizzati per i motivi per cui sono stati raccolti e divugati solo a terze parti autorizzate), la possibilità per l'utente di poter scegliere se le sue informazioni devono essere raccolte per farne uso all'interno del cloud e un sistema di notifiche agli utenti su quali dati saranno raccolti, come verranno utilizzati e per quanto tempo essi saranno salvati all'interno del cloud.

L'aspetto più importante dell'architettura proposta è il *Privacy Manager*, (Figura 2.4), un software nella parte client che aiuta l'utente a proteggere la sua privacy quando accede al cloud. Una sua caratteristica fondamentale è la possibilità di poter offuscare e deoffuscare le informazioni sensibili mentenute nel cloud.

Inoltre, permette all'end-user di esprimere alcune preferenze sul trattamento dei dati sinsibili, utilizzare più profili per lo stesso utente, rivedere e correggere le informazini salvate nel cloud. Un analisi delle feature più importanti di questo software:

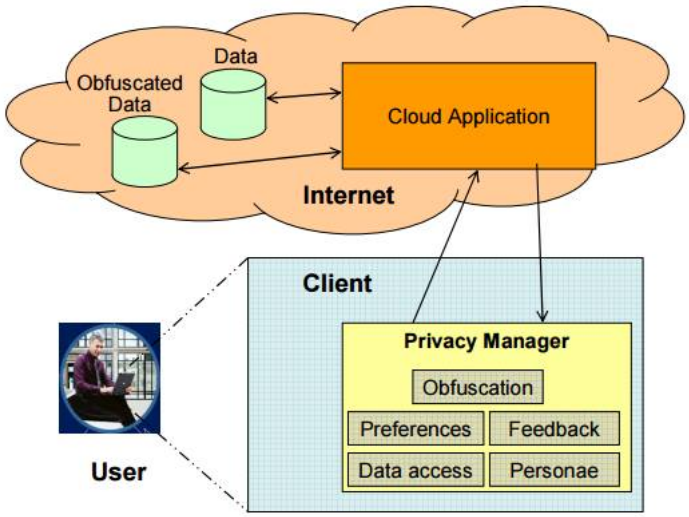

Figura 2.4: Struttura del Privacy Manager

- *Offuscamento dei dati.* Questo strumento può automaticamente offuscare alcuni o tutti i campi di una struttura dati, prima che sia inviata al cloud per essere processata, e de-offuscarli una volta estratti dal cloud. Queste operazioni vengono eseguite utilizzando una chiave scelta dall'utente e non rivelata al cloud service provider, non permettendo così alle applicazioni sul cloud di poterne vedere il contenuto. Inoltre, poiché la chiave è unica, nessun "attaccante", pur utilizzando la stessa applicazione, è in grado di deoffuscare i dati. In generale, più informazioni vengono offuscate dentro una struttura dati, minore è il numero di applicazioni che può utilizzare quei dati come input e più lento è il processo di offuscamento. In alcuni casi, si può scegliere di non offuscare tutte le informazioni personali, poiché l'utente potrebbe scegliere di mantenerne qualcuna in chiaro in modo da ricevere dei contenuti personalizzati (p.e. pubblicità).

- *Settaggio delle preferenze.* Questo metodo permette agli utenti di impostare le loro preferenze sulla gestione dei dati salvati sul cloud e che non sono offuscati. Le scelte effettuate saranno poi associate ai dati inviati al cloud. Una parte di queste opzioni può implicare anche lo scopo per cui i dati possono essere usati nel cloud.

- *Accesso ai dati.* Il privacy Manager contiene un modulo che permette agli utenti di accedere alle informazioni personali nel cloud, così da controllare quali dati siano stati trattenuti e poterne controllare l'accuratezza. Questa feature, però, è un meccanismo di verifica e rilevazione di eventuali violazioni di privacy solo dopo essere avvenute, e non agisce per prevenzione delle stesse. Questo perché, nonostante i principi base di accesso ai dati e accuratezza degli stessi siano considerati privacy in molte leggi nazionali di settore, i service provider hanno la "necessità" di rendere queste informazioni accessibili agli utenti. Questa feature abilita, organizza e registra questi accessi sulla macchina del cliente. La possibilità di fornire accesso ai dati, quando essi sono sparsi su un largo numero di macchine, è un problema molto arduo.

- *Feedback.* Questo modulo gestisce e visualizza i feedback all'utente, e implica un sistema di notifica all'utente a seconda dell'uso che si fa delle sue informazioni personali. Inoltre, può monitorare i dati personali che vengono trasferiti dalla piattaforma e potrebbe avere anche un ruolo educativo sulle questioni di privacy e aiutare l'utente ad attuare scelte consapevoli, oltre a quelle espresse nel settaggio delle preferenze.

- *Profili multipli.* Questa feature permette agli utenti di scegliere tra diversi profili quando devono interagire con il cloud. Per esempio, in alcuni contesti l'utente potrebbe non voler condividere informazioni personali e di conseguenza agire in modalità anonima. La scelta da parte dell'utente di quale profilo utilizzare potrebbe "guidare" il livello di offuscamento utilizzato. Ad esempio, all'interno di un insieme di dati si potrebbe voler offuscare delle informazioni piuttosto che altre a seconda del profilo utilizzato.

Un esempio del processo di offuscamento viene rappresentato in figura 2.5. Un rappresentate di un'azienda, che utilizza il privacy manager, vuole cercare gli indirizzi email dei clienti che hanno speso di più per un prodotto CoolWidget. La query prodotta dal rappresentante viene offuscata e inviata ad un'applicazione del service provider eseguita sul cloud. L'applicazione consulta il database offuscato delle vendite e invia una risposta anch'essa offuscata. Perciò il software de-offusca i dati e restituisce gli indirizzi email.

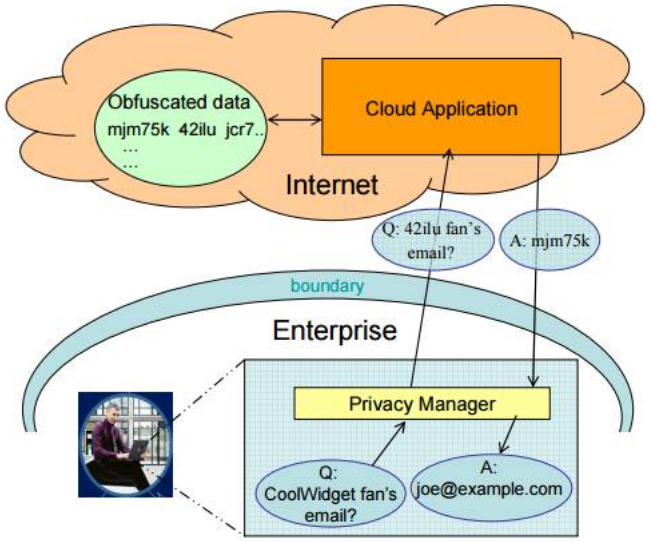

Figura 2.5: Processo di offuscamento

La scelta di utilizzare l'offuscamento avviene poiché i dati, con questo tipo di tecnica, trattengono qualche informazione di quelli originari. Infatti potrebbe essere possibile, per qualche tipo di informazione come le vendite, ottenere delle analisi direttamente dai dati offuscati. Per esempio, con il metodo descritto precedentemente, sarebbe possibile sapere quale prodotto è stato maggiormente venduto. Tuttavia, si potrebbero anche utilizzare tecniche di offuscamento più complesse che non permetterebbero tali operazioni. Sono stati compiuti diversi studi sull'uso della crittografia per quanto riguarda i dati [24], tuttavia tale argomento non verrà trattato in questa tesi poiché aprirebbe a nuovi scenari come le difficoltà di processare e indicizzare dati criptati e le tecniche ad esso correlate.

Tali argomenti ritengo che potranno essere trattati come sviluppi futuri di questa tesi.

Capitolo 3

Sistemi di verifica

In questo capitolo verrà discusso il metodo di valutazione del livello di trust. Tale questione è specificata "sistema di misurazione dell'affidabilità" (trustworthiness measuring system), il quale determina il livello di trust di un sistema cloud o, in generale, di una Service-Oriented Architecture con l'ausilio di una ben definita scala di affidabilità [22]. Considerando i diversi agenti (service provider, utenti, consumatori, produttori etc.) coinvolti in questo contesto, nella prossima sezione verrà affrontato il problema di un sistema di fiducia tra di essi.

3.1 L'ambiguità del concetto di trust

Misura e previsione della trustworthiness non sono operazioni semplici, a causa della natura ambigua, dinamica e complessa del concetto di trust. In particolare, la natura ambigua deriva dal fatto che il concetto di trust è di per sé qualcosa di indefinito e impreciso, e quando si prova a darne la definizione oppure a spiegare un livello di fiducia esso risulta molto vago. L'aspetto dinamico è dovuto al fatto che il valore di trust non può essere stabile ma cambia col passare del tempo. Il fatto che ci siano molteplici modi per determinarla e una varietà di visioni su di essa ne determina la complessità. In generale, quando qualcosa non può essere definito in modo chiaro, non è stabile e costante, e sono presenti una varietà di visioni e opinioni, è sempre difficile gestirne e prevederne i futuri valori.

Nel concetto di trust vengono identificate sei importanti peculiarità ambigue e dinamiche, importanti per comprendere meglio la sua complessità e i metodi di misurazione e previsione. Una di queste è il fatto che la fiducia è **implicita**, cioè un agente che deve effettuare una valutazione di affidabilità non può essere capace di articolare esplicitamente e specificare la convinzione, la disponibilità, il meccanismo, il contesto e la dipendenza temporale della fiducia espressa. La ragione di ciò sta nella presenza di molti elementi che influiscono nell'instaurazione di un rapporto di fiducia. Alcuni di questi sono chiaramente percepibili, mentre altri non sono semplici da differenziare e devono essere costruiti lentamente e in modo incrementale attraverso l'esperienza data dal rapporto. L'**asimmetria** o non mutua reciprocità della fiducia è un'altra caretteristica portatrice di ambiguità. Per asimmetria si intende uno scenario in cui un agente che deve effettuare valutazioni di fiducia ha una certa convinzione in un altro agente, il ché però non implica anche la fiducia in senso opposto e nel medesimo contesto. Anche la misurazione e previsione di trust sono asimmetriche e un valore di fiducia non rappresenta entrambe le parti in un rapporto. Una situazione di **transitività** invece si manifesta quando un agente A si fida di un agente B e quindi automaticamente di ogni agente affidabile per B. Questa, conosciuta anche come "fiducia derivata", è il risultato di un rapporto tra agenti già esistente, deve essere considerata all'interno dello stesso contesto ed è veramente difficile da quantificare accuratamente. La transitività è un concetto importante in uno scenario in cui utenti o agenti anonimi vogliono identificare la qualità del servizio attraverso un'introduzione transitiva (raccomandazione), quest'ultima considerata però ambigua poiché deve essere contestualizzata e inserita in un preciso arco temporale, informazioni che non risultano sempre esplicite. Il concetto di **contrarietà** è relativo al contesto nel quale la fiducia viene applicata, poiché esso può essere inteso in modo diverso dagli agenti che vi partecipano e quello che può risultare chiaro per uno potrebbe non esserlo per l'altro. L'ideale, per effettuare delle valutazioni o previsioni di trust, è definire dei contesti più chiaro possibile, in caso contrario la valutazione dell'affidabilità non ha valore, poiché il contesto influenza l'abilità di valutazione. **L'asincronia** del concetto di trust si riferisce al lasso di tempo in cui essa viene

valutata, poiché questo arco temporale può essere supposto o definito in maniera diversa dagli agenti coinvolti, e questo porta ad ambiguità e a difficoltà ad effettuare valutazioni di affidabilità. Di conseguenza, l'ideale sarebbe poter aggregare una media delle valutazioni di più archi temporali, in modo da avere una visione complessiva dell'affidabilità in tempo. Infine, la **serietà** dei rapporti di fiducia, percepita da ogni parte coinvolta, è l'ultima delle caratteristiche che portano ad ambiguità. Possibili scenari potrebbero comportare la presenza di due agenti A e B in cui da una parte il rapporto viene considerato di una certa importanza, mentre dall'altra no.

Considerate queste criticità del concetto di trust, va detto che esistono anche dei meccanismi per la valutazione dei service provider. Una parte di questa operazione dipende dalla misurazione della qualità del servizio, considerata uno dei compiti più difficili, poiché bisogna considerare sia i giudizi espressi dall'utente sia quelli del provider. In seguito verrà approfondito questo aspetto con l'utilizzo del metodo CCCI.

3.2 Affidabilità

Nella letteratura esistente, non si nota un vero e proprio sistema metodico e rigoroso su come determinare il livello di affidabilità di un agente, né tantomeno una scala standardizzata che possa rappresentare il livello di trust. Quando si parla di *trustworthiness*, si intende una misura del livello di trust che un agente trusting (colui che dà la fiducia) ha nei confronti di un altro agente trusted (colui che deve ottenere la fiducia). L'affidabilità viene quindi misurata attraverso scale di trustworthiness, le quali rappresentano i sistemi di riferimento per tali misurazioni.

Per fare un po' di chiarezza con la terminologia, quando si parla di "misura" si intende una stima del livello o del grado di fiducia. Una stima è il risultato di un tentativo di misura, il quale può essere l'opinione o la valutazione di un esperto oppure un giudizio o previsione scientifici. Una misura dà una stima approssimata, a differenza di un metodo che fa uso di una scala standardizzata, e spesso il risultato è un importo o un valore.

Con "livello di trust" si fa riferimento alla "quantità" di fiducia che un agente trusting ha nei confronti di uno trusted, e può essere rappresentato anche in modo non numerico. Questa caratteristica è **unidirezionale**, se l'agente trusting ha un alto grado di fiducia, questo implica che il livello di affidabilità, su una scala trustworthiness, del trusted agent è alto. Il livello di trust descrive un singolo scenario e muta di volta in volta.

Trustworthiness scale (ordinal scale)	Semantics (linguistic definitions)	Percentage intervals (user defined) (%)	Trustworthiness value (user defined)
Level −1	Unknown agent	n/a	$x = -1$
Level 0	Very untrustworthy	0–19	$x = 0$
Level 1	Untrustworthy	20–39	$0 < x \leq 1$
Level 2	Partially trustworthy	40–59	$1 < x \leq 2$
Level 3	Largely trustworthy	60–79	$2 < x \leq 3$
Level 4	Trustworthy	80–90	$3 < x \leq 4$
Level 5	Very trustworthy	90–100	$4 < x \leq 5$

Figura 3.1: Scala di trust a sette livelli.

Una "trustworthiness scale", proposta in [22], consiste in un semplice metodo che aiuta a determinare il grado di fiducia tra due agenti, e anch'essa può non essere visualizzata in un formato numerico. Una scala non numerica può rappresentare una classificazione di fiducia per mezzo di livelli o gradi di accertamento, esprimendoli con termini di categoria come "molto affidabile" o dando classificazioni come le classiche cinque stelle. Analogamente al mondo reale, dove prima di condurre un affare si accerta dell'affidabilità degli altri agenti coinvolti, così nel mondo virtuale utilizzare una scala di affidabilità è un'importante pratica per il trusting agent, il quale necessita di uno strumento tale prima di poter intraprendere qualsiasi affare. Tuttavia, in questo scenario, dove le transazioni vengono eseguite tra agenti mai incontrati prima (fisicamente), i pericoli sono probabilmente più elevati. Inoltre, molti siti e piattaforme, come Amazon, E-Bay e Yahoo, utilizzano sistemi di classificazione di trust che automaticamente assegnano affidabilità a tutti gli agenti (sia provider che clienti) con i quali interagiscono.

Sono stati compiuti diversi studi su quanti livelli di trust sia giusto utilizzare per avere una scala di affidabilità efficiente ([26] e [27]). Da un punto di vista pratico, cinque sembrano essere pochi mentre dieci risulterebbero non necessari. Considerando una scala con 10 possibili livelli, verrebbero create delle valutazioni come 'molto inaffidabile', 'inaffidabile', 'quasi affidabile' e "minimamente affidabile" utili in uno scenario di tipo sociale, ma che presenterebbero delle distinzioni praticamente insignificanti all'interno di un contesto commerciale, poiché nessun agente vorrebbe avere un'interazione con un provider con tale classificazione. Da qui in poi verrà presa in esame una scala di affidabilità a sette livelli. In genere, è preferibile utilizzare un approccio numerico, o perlomeno misto (come descritto in figura 3.1), in modo da evitare l'ambiguità di descrizioni espresse con il linguaggio naturale.

Trustworthiness level	Trustworthiness value (user defined)	Visual representation (star rating system)	QoS rating (linguistic definitions)
Level −1	$x = -1$	Not displayed	New agent
Level 0	$x = 0$	Not displayed	Terrible
Level 1	$0 < x \leq 1$	From ★ to ⯪	Bad
Level 2	$1 < x \leq 2$	From ★⯪ to ★★	Average
Level 3	$2 < x \leq 3$	From ★★⯪ to ★★★	Good
Level 4	$3 < x \leq 4$	From ★★★⯪ to ★★★★	Very good
Level 5	$4 < x \leq 5$	From ★★★★⯪ to ★★★★★	Super or excellent

Figura 3.2: Scala di trust a sette livelli con classificazione di QoS

Come evidenziato in figura 3.2, i sette livelli di affidabilità possono essere rappresentati anche per mezzo delle classiche 'stelle'. Sempre dalla stessa immagine si può notare come i livelli 0 e -1 siano contrassegnati come "not displayed" poiché i clienti difficilmente sarebbero interessati ad eseguire transazioni con provider che devono essere ancora classificati. D'altro canto, anche dal punto di vista di un provider, sarebbe un inutile spreco di tempo fornire dei servizi che risulterebbero sconosciuti o non affidabili, poiché gli utenti in ogni caso non usufruirebbero di quel servizio. Lo scopo di un sistema di misurazione dell'affidabilità è presentare

la qualità del servizio (QoS) o quella di un cloud provider e aiutare gli utenti a trovare dei sistemi che possano andare incontro alle proprie necessità, oltre ad aiutare i provider a costruire la propria reputazione. Inoltre, si può notare come i sette livelli vengano divisi essenzialmente in trust negativi e trust positivi, considerata una descrizione addizionale per chiarire meglio i gradi di affidabilità. Un trust negativo è dato ad un agente che non effettua un buon servizio o non si comporta secondo gli accordi presi con il trusting agent. In questo caso, il trust negativo viene rappresentato con affidabilità minore di 1, poiché descriverlo in modo più dettagliato non avrebbe comportato una differenza significante nel contesto virtuale. Al contrario, un trust positivo viene assegnato ad un agente che si appresta ad avere o possiede già le capacità per soddisfare anche solo una parte dei servizi che erano stati concordati con il trusting agent. Per andare più nel dettaglio, le sette classificazioni di affidabilità possono essere descritte come segue:

- **Livello -1:** ***affidabilità sconosciuta***. Classificazione assegnata quando non si è in grado di accertare o stimare il livello di fiducia di un trusted agent, in un dato contesto e periodo di tempo. Questo può essere dovuto a diversi fattori: (i) Il trusted agent è al suo primo accesso nella rete; (ii) il trusting agent non ha avuto precedenti interazioni con esso; (iii) tutti gli altri agenti affidabili, "interrogati" per ottenere raccomandazioni sul trusted agent, non hanno mai avuto interazioni con esso e non hanno alcuna informazione sulla sua affidabilità; di conseguenza, il trusting agent non può prendere nessuna decisione e assegna il valore -1. Tale classificazione non implica nessun tipo di diffidenza o sospetto, poiché tutti i nuovi agenti hanno un livello iniziale pari a -1. Questo tipo di affidabilità generalmente non viene visualizzato, poiché è molto usuale tra gli utenti di Internet e i service provider, considerando anche che si è interessati a notificare quei service provider che siano in grado di fornire i servizi richiesti.

- **Livello 0:** ***molto inaffidabile***. Assegnato quando il trusting agent non è affidabile in uno specifico contesto e in un particolare periodo di tempo, perciò è preferibile che esso non esegua determinate operazioni. Tale situazione

può essere dovuta al fatto che la maggior parte o tutti gli agenti interrogati per fornire una raccomandazione hanno assegnato un valore di affidabilità pari a 0, oppure il trusted è stato coinvolto in un accordo fraudolento con l'agente trusting in cui quest'ultimo aveva comunicato in termini chiari ed espliciti l'intera metodologia di valutazione della fiducia del trusted. Questo rappresenta il più alto livello possibile di trust negativo. Ciò non significa che il trusted agent non possa, in future interazioni, comportarsi in modo opposto alla valutazione assegnatagli. Anche questo grado di trust, per gli stessi motivi del punto precedente, non viene visualizzato. Tuttavia potrebbe essere utile, ad esempio per il sito di un'azienda di commercio, avere tale informazione nel caso in cui un provider non fidato o utenti non affidabili chiedano di essere inseriti sul sito dell'azienda, o un altro utente li suggerisca come affidabili.

- **Livello 1: *inaffidabile***. Classificazione che viene data quando si ha molta poca fiducia nel trusted agent, considerato uno specifico contesto in un determinato arco temporale. Viene data una valutazione $0 < x \leq 1$, a causa dell'assegnazione, da parte della maggior parte degli agenti interrogati, di un livello di affidabilità pari a 1 oppure perché il trusted agent non è in grado di offrire il servizio commissionato o di soddisfare i termini del contratto stipulato con il trusting agent, il ché potrebbe implicare anche un inganno subito da quest'ultimo. Questa valutazione rappresenta l'inizio della classificazione 'negative trust'. Come nei casi precedenti, anche in questo contesto il trusting agent potrebbe in futuro comportarsi in modo diverso rispetto alla valutazione datagli, e molti servizi non visualizzano i provider o gli utenti di questo livello.

- **Livello 2: *parzialmente affidabile***. Valutazione data quando si ha circa il 50% di fiducia nel trusted agent, in un dato contesto e in un lasso di tempo limitato. Questo livello indica che l'affidabilità dell'agente è ancora incerta, tuttavia marca l'inizio del trust positivo. L'assegnamento di tale livello ad un trusted agent indica che il suo comportamento non era meritevole di una

classificazione negativa, ciò nonostante esso non agisce in maniera coerente nel rispetto delle attese del trusting agent, poiché potrebbe non garantire la qualità del servizio in modo continuativo. La valutazione data in questo caso è $1 < x \leq 2$, a causa degli agenti interrogati sull'affidabilità di colui che deve essere valutato, perché il suo comportamento non può essere catalogato né come cattivo nè come buono oppure perché ha fornito un buon servizio solo per metà del tempo di valutazione.

- **Livello 3: *fondamentalmente affidabile***. In questo caso viene assegnato un valore $2 < x \leq 3$ e il trusted agent viene considerato affidabile al 70%. Questo può dipendere dagli altri agenti coinvolti oppure dal fatto che il valutato ha agito in modo giusto per la maggior parte del tempo preso in esame e in uno specifico contesto. L'agente valutante in questo caso si aspetta che il comprimario si comporterà bene nelle future interazioni, e generalmente questo avviene. Tuttavia potrebbero capitare delle occasioni in cui esso non rispetta pienamente le aspettative.

- **Livello 4: *affidabile***. Un agente viene considerato 'affidabile' quando gode di un quoziente di fiducia oltre l'80%. Questo significa che l'agente valutato, soddisfacendo i criteri, ha rispettato molte delle aspettative del trusting agent. In questo caso viene data una valutazione $3 < x \leq 4$ che rappresenta a tutti gli effetti un trust positivo e, in future possibili interazione, il trusting agent avrà un'alta considerazione dell'agente valutato.

- **Livello 5: *molto affidabile***. Questo rappresenta il più alto livello di fiducia assegnabile ed indica un affidabilità oltre il 90%. Tale valutazione deriva dal fatto che il trusted agent si è sempre comportato esattamente come da aspettative, ed è garanzia dei suoi futuri atteggiamenti. In questo scenario viene data una valutazione di $4 < x \leq 5$.

Le misurazioni di affidabilità avvengono in seguito ad interazioni dirette. Date una serie di interazioni del passato, si forniscono una serie di valutazioni trust. Solo una volta ottenuti un insieme di valori precedenti si è in grado di esporre

una previsione di affidabilità. Il valore di trust può essere riferito ad un agente, un prodotto o un servizio. La misurazione dà un'idea se è il caso di interagire nuovamente con il trusted agent, ne determina gli attributi di fiducia e affidabilità, specifica gli attributi e l'affidabilità di una relazione fidata.

Per quanto riguarda la qualità del QoS, ci sono tre tipi di misure: reputazionale, storica e per interazione diretta. Tuttavia, solo quest'ultima risulta in una valutazione di affidabilità, mentre la reputazione fornisce opinioni di terze parti che non sono del trusting agent e la storia esamina i valori di trust passati.

Al contrario, le previsioni di trust avvengono prima di ogni interazione con l'agente trusted, e in tal caso colui che valuta si deve basare su dati storici o reputazionali. La valutazione può essere riassegnata dopo un'interazione diretta. I valori di trust e affidabilità vengono usati solamente da colui che deve accertare la fiducia, e non devono necessariamente essere condivisi con gli altri agenti. Nel caso in cui le misure o previsione di trust siano sbagliate, il trusting agent è l'unico ad averne la responsabilità. Tuttavia, se decide di rendere pubbliche delle informazioni sbagliate, pur non avendo nulla da perdere, rischia di mettere il trusted agent in una posizione critica, perciò è di fondamentale importanza che il trusting agent sia onesto, imparziale e sincero.

3.3 Misurazioni di affidabilità

Nelle sezioni precedenti è stato evidenziato il contesto al quale il concetto di trust si riferisce e la sua natura dinamica e complessa. In questo paragrafo verrà analizzato un metodo per la misurazione effettiva della fiducia, che comprende la qualità degli agenti coinvolti e dei servizi in modo da migliorare la sicurezza dell'utente. Per fare ciò, bisogna definire dei criteri per la valutazione della qualità, tenendo in conto la chiarezza e trasparenza di ognuno di essi, l'influenza che esercitano sulla valutazione e la loro correlazione con la qualità del servizio.

Lo scopo delle misurazioni di affidabilità è constatare la *qualità* del trusted agent. Quanto si parla di qualità o trust in riferimento a qualcosa, normalmente il contesto e già definito, ma questo da solo non fornisce abbastanza informa-

zioni per giustificare la valutazione assegnatagli. In generale, si può dire che una metodologia di misurazione di trustworthiness contiene quattro passaggi principali:

1. Focalizzazione sul **contesto** e la conoscenza del suo **dominio**. Un contesto definisce uno specifico dominio, marcandone il nome, e include funzioni distinte. Può essere decomposto in criteri disposti in modo gerarchico. Un contesto rappresenta l'agente, il servizio, il prodotto o le loro funzioni ed è composto da tre parti: nome, tipo e descrizione, che diventeranno poi caratteristiche del servizio. Un contesto può rappresentare un dominio fisico, inteso come agente, servizio o prodotto, oppure un dominio concettuale che rappresenta astrazioni come team virtuali o operazioni gestionali. Per misurare la qualità in un contesto fisico, esso deve essere diviso in diversi aspetti qualitativi.

2. Identificazione degli **aspetti qualitativi** derivanti dalla conoscenza del dominio e del contesto. In particolare, tali aspetti servono a definire il 'perché' delle misurazioni. Se questa fase risulta difficoltosa, come approccio alternativo vengono identificati il 'service-level agreement' o il contratto, i quali pur non specificando chiaramente gli aspetti di qualità, forniscono qualche indicazione su di essi, fissando i comportamenti che il trusted agent deve rispettare e le aspettative del trusting agent.

3. Sviluppo di **criteri per la valutazione** della qualità per ogni suddetto aspetto, in modo da avere una rappresentazione numerica o un sistema di classificazione della stessa. Questi criteri saranno usati per definire le metriche di misurazione della qualità. Una metrica è definita come "un'insieme di condizioni o dati con le quali ogni aspetto qualitativo viene confrontato, correlato, quantificato e giudicato". I criteri di valutazione agiscono come linee guida da seguire per il trusted agent. Nel caso in cui essi non vengano ben compresi da entrambe le parti, la valutazione di affidabilità fallirebbe, con responsabilità da condividere tra gli agenti coinvolti. Se il trusting agent effettua una misurazione che intende utilizzare in futuro, è suo dovere fare in modo che l'accordo e i criteri siano ben compresi, altrimenti creerà le condi-

zioni ottimali per attività fraudolente, oltre agli errori che ne conseguiranno al momento della valutazione.

4. Misurare la correlazione della qualità dei servizi nei confronti dei criteri scelti per la valutazione attraverso le metriche CCCI. In questo passaggio bisogna notare che il peso di ogni criterio può essere differente. Tuttavia, i criteri possono essere non ben compresi da uno o entrambi gli agenti, e possono evolvere nel tempo, quindi è importate definirli chiaramente.

3.4 Metodologia CCCI

Le metriche sono specificatamente designate per la misurazione della qualità degli agenti e per quella del servizio offerto (QoS). In particolare, questa metodologia è composta da quattro metriche chiave citate precedentemente e che ora saranno analizzate più nello specifico.

3.4.1 Correlazione delle qualità definite

Questa tecnica è largamente utilizzata negli studi statistici, la sua funzione primaria è studiare il gap (distanza) e le differenze tra due entità (oggetti, classi, variabili, etc.) per un dato periodo di tempo e monitorare le variazioni inerenti, per scopi di verifica, stima o previsione. Quando si effettua una correlazione per misurare o stimare l'affidabilità, si presentano almeno due elementi comparabili: uno può essere usato come criterio o variabile e l'altro per mapparlo. L'impegno iniziale di un provider può essere utilizzato come metro di paragone per gli attuali servizi offerti. La correlazione viene rappresentata con $\mathbf{Corr_{qualities}}$, dove 'Corr' è la metrica per la misura della chiarezza, 'qualities' rappresenta la qualità offerta dal trusted agent in modo che tutti i criteri per la valutazione della qualità siano usati per la valutazione di trustworthiness, mentre 'c' si riferisce al particolare criterio, e può essere determinata da: (i) un confronto tra la qualità definita originariamente e l'attuale QoS corrispondente ad ogni criterio che il trusting agent

sta cercando nell'interazione; (ii) l'accumulazione di tutti i valori di correlazione per tutta l'interazione.

La metrica può essere espressa come la somma dei valori di correlazione per tutti i criteri definiti nell'interazione. Assumendo che ci siano **N** criteri e che $\mathbf{Commit_{criterion\ c}}$ denoti il compimento del 'c-esimo' criterio, il contributo di questo criterio al valore complessivo di $\mathrm{Corr_{qualities}}$ può essere rappresentato con l'equazione espressa in figura 3.3, oppure con la seguente formula:

$$\boldsymbol{f}(\mathbf{Commit_{criterion\ c}}, \mathbf{Clear_{criterion\ c}}, \mathbf{Inf_{criterion\ c}}) = \mathbf{Commit_{criterion\ c}} * \mathbf{Clear_{criterion\ c}} * \mathbf{Inf_{criterion\ c}}$$

Per determinare il valore di correlazione per un'interazione, il trusting agent deve quindi considerare:

- I criteri utilizzati per determinare l'affidabilità del trusted agent;
- Il valore di $\mathrm{Commit_{criterion}}$ per ogni criterio;
- La somma di tutti i $\mathrm{Commit_{criterion}}$ adegutamente pesati con $\mathrm{Clear_{criterion}}$ e $\mathrm{Inf_{criterion}}$, in un'interazione per ottenere $\mathrm{Corr_{criterion}}$.

In aggiunta, $\mathrm{Commit_{criterion}}$ dovrebbe essere valutato da:

- Se il criterio come base di assegnamento di trustworthiness è stato comunicato chiaramente al trusted agent ($\mathrm{Clear_{criterion}}$);
- l'influenza di ogni criterio nell'interazione ($\mathrm{Inf_{criterion}}$) dal punto di vista del trusting agent.

Alcuni termini utilizzati poc'anzi verranno analizzati meglio in seguito.

$\mathrm{Corr_{qualities}}$ fornisce un metodo su come il valore di correlazione di un servizio può essere ottenuto utilizzando i livelli di trustworthiness, una volta che il trusting agent ha determinato i criteri sui quali assegnare l'affidabilità al trusted agent. Se un particolare criterio non è stato espresso chiaramente al trusted agent

($\text{Clear}_{\text{criterion}}$ con valore 0), non verrà tenuto in conto nel calcolo dell'affidabilità. D'altro canto, se il valore di $\text{Clear}_{\text{criterion}}$ sarà pari a 1, questo verrà tenuto in considerazione, per questo motivo il valore di $\text{Commit}_{\text{criterion}}$ viene moltiplicato per quello di $\text{Clear}_{\text{criterion}}$. Allo stesso modo viene concepito il calcolo legato all'influenza ($\text{Inf}_{\text{criterion}}$).

$$\text{Corr}_{\text{qualities}} = \sum_{C=1}^{N} f(\text{Commit}_{\text{criterion}\,c}, \text{Clear}_{\text{criterion}\,c}, \text{Inf}_{\text{criterion}\,c})$$
$$= \sum_{C=1}^{N} \text{Commit}_{\text{criterion}\,c} * \text{Clear}_{\text{criterion}\,c} * \text{Inf}_{\text{criterion}\,c}$$

Figura 3.3: $\text{Corr}_{\text{qualities}}$ per un dato servizio espresso come somma.

3.4.2 Dedizione al criterio

Questa metodologia misura 'quanto' l'aspetto qualitativo offerto dal trusted agent è conforme al criterio di valutazione definito. Il compimento di ogni impegno è una misura di quanto il trusted agent abbia rispettato gli accordi presi originariamente. Il livello di 'dedizione' per il criterio di qualità è rappresentato da $\mathbf{Commit_{criterion\ c}}$, dove 'Commit' indica il metodo di misurazione del livello di impegno per il criterio di qualità, 'criterion' si riferisce ai criteri di valutazione della qualità e 'c' al particolare criterio. In un'interazione con il trusted agent, colui che effettua la valutazione deve considerare la correlazione tra la QoS definita in principio e quella offerta al momento, effettuando un confronto con ogni criterio dell'interazione. La correlazione deve essere definita come un valore numerico che quantifica l'entità del confronto tra l'impegno preso originariamente dal trusted agent, per quel determinato criterio, e la qualità del servizio offerta al momento. Se un servizio è composto da un numero di criteri, bisogna valutare il servizio per ogni singolo criterio, e questo viene rappresentato così:

$$\mathbf{Commit_{criteria}} = \mathbf{Commit_{criterion\ 1}} + \mathbf{Commit_{criterion\ 2}} + \cdots + \mathbf{Commit_{criterion\ N}}$$

Considerando il modello preso in esame in precedenza, con i sette livelli di trustworthiness, è possibile creare una scala con altrettanti livelli di "dedizione" facendoli combaciare con quelli di affidabilità. Il calcolo viene eseguito sia con valori che con livelli.

3.4.3 Chiarezza del criterio

Come detto in precedenza, è importante definire ogni criterio in modo chiaro al momento di determinare l'affidabilità di un agente. Gli aspetti qualitativi e i criteri di valutazione devono essere specificati in modo chiaro, compresi da ambo le parti e mutualmente accettati. Se ciò non avvenisse, le valutazioni di affidabilità non potrebbero essere eseguite correttamente e ne potrebbero conseguirne delle dispute. Se una delle due parti non è convinta della chiarezza dei criteri di valutazione, allora l'interazione tra di loro deve essere ritardata fin quando non si raggiungerà un chiarimento. Questa caretteristica è rappresentata da $\mathbf{Clear_{criterion\ c}}$. La chiarezza di un criterio esprime se o no esso viene compreso in modo chiaro da ambo le parti ma soprattutto da parte del trusted agent, poiché è la sua affidabilità quella che deve essere valutata. Ogni criterio deve essere ben definito prima di 'firmare' il service agreement, così la QoS definita viene comunicata e ogni criterio viene espresso esplicitamente, compreso da entrambe le parti e mutualmente accettato. Ogni criterio sarà usato per la valutazione dell'affidabilità del trusted agent e dovrà essere comunicato ad ambo le parti in modo non ambiguo.

Normalmente, l'utente e il provider devono stipulare un service agreement. In molte situazioni, questo viene trattato come insignificante o avere un liguaggio troppo tecnico e quindi può non essere compreso dall'utente. Molti utenti semplicemente si fidano dell'alto livello di spiegazioni del provider e leggono solamente dei concetti chiave dei loro contratti, tuttavia spesso si possono riscontrare costi, condizioni o asserzioni nascoste in qualche codice non facilmente visibile. La chiarezza di ogni criterio viene valutata con:

$$\mathbf{Commit_{criterion\ 1} * Clear_{criterion\ 1} + Commit_{criterion\ 2} * Clear_{criterion\ 2} + . . . + Commit_{criterion\ N} * Clear_{criterion\ N}}$$

Anche in questo caso si può effettuare una rappresentazione con sette livelli di chiarezza, confrontandoli con quelli di trustworthiness.

3.4.4 Influenza di un criterio

È importante ponderare bene l'influenza di un criterio all'atto di determinare l'affidabilità, data la presenza di diversi criteri e considerando il fatto che alcuni sono più importanti di altri. L'influenza di ogni impegno è definito come una misura di 'come' sia importante il criterio al momento di decidere l'affidabilità del trusted agent. L'influenza viene rappresentata con $\mathbf{Influence_{criterion\ c}}$, mentre quella di un singolo criterio viene indicata con $\mathbf{Inf_{criterion}}$, dove 'Influence' è la metrica utilizzata per la misura dell'influenza, 'criterion' è il criterio e 'c' rappresenta il criterio specifico. L'importanza di un criterio, in un'interazione, è determinata sulla base delle percezioni di colui che deve valutare l'affidabilità e può variare: (i) a seconda della prospettiva del trusting agent, il quale può considerare un'interazione come fallimentare se determinati criteri non vengono soddisfatti; (ii) considerando il fatto che un trusting agent può avere una percezione completamente differente rispetto ad un altro agente col suo stesso ruolo. Se un servizio contiene una serie di criteri, bisogna "pesare" ogni criterio poiché ognuno di essi può avere un differente impatto sul valore di trustworthiness, questo viene rappresentato con:

$$\mathbf{Commit_{criteria\ 1} * Clear_{criterion\ 1} * Inf_{criterion\ 1} + Commit_{criterion\ 2} * Clear_{criterion\ 2} * Inf\ _{criterion\ 2} + . . . + Commit_{criterion\ N} * Clear_{criterion\ N} * Inf_{criterion\ N}}$$

Anche in questo caso è possibile riutilizzare il modello dei 7 livelli, in modo da poter rappresentare in ordine crescente l'importanza dei criteri. L'influenza di ogni criterio all'atto di determinare l'esito di un'interazione sarà presa in considerazione per determinare l'affidabilità del trusted agent. L'influenza di ogni criterio dovrebbe essere comunicata in termini chiari, e soprattutto prima che l'interazione abbia inizio.

Conclusioni

In questo lavoro di tesi è stata effettuata una panoramica dei principali problemi di privacy, trust e sicurezza nel contesto del cloud computing. In particolare, è stata prima svolta un'analisi delle criticità provenienti da altri contesti, e in seguito sono stati presentati i problemi emergenti e quelli che probabilmente compariranno nei prossimi anni.

In seguito, sono stati presentati i passi fatti da alcune aziende nella direzione di sistemi più sicuri e fidati, e alcuni modelli cloud, presenti in letteratura, con alcuni strumenti utili per salvaguardare la privacy dei dati e altri aspetti inerenti. In particolare, si è eseguita un'analisi il più possibile approfondita sull'infrastruttura di Amazon e su come i suoi servizi vengono offerti.

Infine, si è cercato di fare chiarezza sul concetto di trust nelle Service-Oriented Architecture (SOA) e sugli elementi ad esso connessi, presentando alcuni sistemi di valutazione per l'affidabilità degli agenti coinvolti nelle possibili interazione che avvengono in tale contesto.

I problemi relativi alla sicurezza, in generale, sono in quantità maggiore rispetto a quelli analizzati in questo lavoro, ed esistono diversi studi che li prendono in esame, provando a classificarne l'origine e la natura, e proponendo possibili soluzioni. Da notare che netti miglioramenti sono stati fatti dalle aziende, con lo scopo di adeguare la tutela della privacy e la sempre più crescente domanda da parte dei consumatori. Tuttavia in alcuni campi, ad esempio per quello che riguarda gli obblighi contrattuali o modi di gestire i dati degli utenti, c'è ancora tanto da fare. Soprattutto alla luce degli eventi degli ultimi anni (p.e raccolta dati da parte di

agenzie governative) che hanno contribuito a creare preoccupazione nei consumatori, la gestione dei dati e la trasparenza della stessa sono dei punti che non vanno trascurati.

I possibili sviluppi futuri di questo lavoro, come già citato a pagina 28, potrebbero riguardare un approfondimento sull'uso della crittografia nei sistemi cloud. In particolare, uno studio potrebbe essere eseguito sulle tecniche utilizzabili per criptare i dati piuttosto che offuscarli, e su come essi possano essere processati e indicizzati. Inoltre, sarebbe interessante sviluppare una valutazione della privacy e del trust per gli ambienti di cloud computing e storage più diffusi, come ad esempio Google App Engine e Dropbox.

Bibliografia

[1] R.Chow, P. Golle, M. Jakobsson, E. Shi, J. Staddon, R. Masuoka, J. Molina. Controlling data in the cloud: outsourcing computation withouth outsourcing control. In *Proceedings of the 2009 ACM workshop on Cloud computing security (CCSW 2009)*, Chicago, IL, USA, November 2009.

[2] VMWare vulnerability.
http://www.securitytracker.com/alerts/2008/Feb/1019493.html.

[3] Google Docs Glitch Exposes Private File.
http://www.pcworld.com/article/160927/google_docs_glitch_exposes_private_files.html.

[4] Salesforce.com Warns Customer of Phishing Scam
www.pcworld.com/businesscenter/article/139353/salesforcecom_warns
_customer_of_phishing_scam.html.

[5] CLOIDIFIN.
http://community.zdnet.co.uk/blog/0,1000000567,2000625196b,00.htm?new_comment

[6] Extended Gmail outage hits Apps admins.
http://www.computerworld.com/article/2533740/web-apps/extended-gmail-outage-hits-apps-admins.html

[7] Amazon S3 Availability Event: July 20, 2008.
http://status.aws.amazon.com/s3-20080720.html

[8] Amazon EC2 Crosses the atlantic.
http://aws.amazon.com/about-aws/whats-new/2008/12/10/amazon-ec2-crosses-the-atlantic/

[9] Organizations urge Google to suspend Gmail.
http://www.privacyright.org/ar/GmailLetter.htm

[10] Xen Vulnerability.
http://vmblog.com/archive/2007/10/02/secunia-reports-xen-vulnerability.aspx#.VXQ6lrw_g7a

[11] VirtualPC vulnerability.
http://www.securitytracker.com/id?1022544.

[12] Cloud Burst as Coghead Calls It Quits. http://www.zdnet.com/article/cloud-bursts-as-coghead-calls-it-quits/.

[13] Loss of costumer data spurs closure of online storage service 'The Linkup'.
http://www.networkworld.com/article/2274737/data-center/loss-of-customer-data-spurs-closure-of-online-storage-service–the-linkup-.html.

[14] Blue Cloud.
https://www-03.ibm.com/press/us/en/pressrelease/26642.wss

[15] Fact Sheet: IBM Rational AppScan

[16] Amazon Web Services: Overview of Security Processes, September 2008.
https://media.amazonwebservices.com/pdf/AWS_Security_Whitepaper.pdf

[17] Sato, H.; Kanai, A.; Tanimoto, S.; A Cloud Trust Model in a Security Aware Cloud. Proceedings of the 2010 10th IEEE/IPSJ International Symposium on Application and the Internet, pp. 121-124, July 19-23, 2010.

[18] M. Mowbray, S. Pearson. A client-Based Privacy Manager for Cloud Computing. Proceedings of the fourth international ICST conference on COMmunication system softWAre and middlewaRE, June 16-19, 2009. Dublin, Ireland.

[19] Ion, Iulia and Sachdeva, Niharika and Kumaraguru, Ponnurangam and Capkun, Srdjan, Home is Safer than the Cloud! Privacy Concerns for Consumer Cloud Storage.Proceedings of the Seventh Symposium on Usable Privacy and Security (SOUPS). (July 14, 2011). Pittsburgh, Pennsylvania.

[20] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia. 2010. A view of cloud computing. Commun. ACM 53, 4 (April 2010), 50-58.

[21] Gary Anthes. 2010. Security in the Cloud. Communication of the ACM, v.53, n.11 (November 2010), 16-18.

[22] E. Chang, T. Dillon, F. K. Hussain. Trust and Reputation for Service-Oriented Environments. Tecnologies for building business intelligence and consumer confidence. John Wiley & Sons. 2005.

[23] T. Ristenpart, E. Tromer, H. Shacham, S. Savage. Hey, You, Get Off of My Cloud! Exploring Information Leakage in Third-Party Compute Clouds. Proceedings of the 16th ACM conference on Computer and communications security, pp. 199-212. November 09-13, 2009, Chicago, Illinois, USA.

[24] C. Cachin, I. Keidar, A. Shraer. Trusting the cloud. ACM SIGACT News, v.40 n.2. June 2009.

[25] Siani Pearson, Taking account of privacy when designing cloud computing services, Proceedings of the 2009 ICSE Workshop on Software Engineering Challenges of Cloud Computing, p.44-52, May 23-23, 2009.

[26] Karl Aberer , Zoran Despotovic, Managing trust in a peer-2-peer information system, Proceedings of the tenth international conference on Information and knowledge management, October 05-10, 2001, Atlanta, Georgia, USA.

[27] A.A. Rahman, S. Hailes (2003). A distributed Trust Model. Proceedings of the 1997 workshop on New security paradigms, p.48-60, September 23-26, 1997, Langdale, Cumbria, United Kingdom.