4u S.r.l.

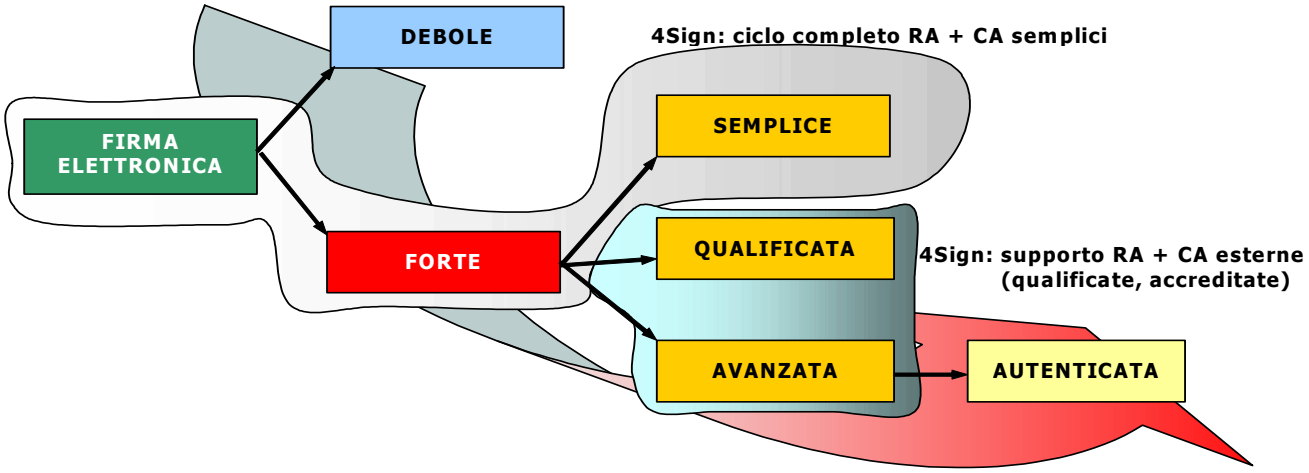
4Sign®

Suite di Firma Digitale
Soluzioni Applicative

4Sign® è una suite software open-source, orientatata a gestire il ciclo completo di firma digitale con certificati X.509, realizzata utilizzando componenti open-source di pubblico dominio e multi-piattaforma, e costituita da un insieme di moduli interoperabili e facilmente integrabili in applicazioni distribuite.

Nel contesto normativo e tecnologico della Firma Elettronica, la suite 4Sign si propone come strumento di gestione della Firma Digitale (Firma Elettronica Forte, realizzata utilizzando la crittografia asimmetrica e i certificati X.509), in grado di:

- coprire completamente il ciclo della Firma Digitale Semplice (supportata da una Registration Authority e da una o più Certification Authority interne all'Azienda),
- supportare completamente l'utilizzo aziendale di certificati qualificati emessi da Enti Certificatori Pubblici (e quindi la Firma Digitale Qualificata e/o Accreditata).



La suite 4Sign comprende quindi:

1. una **Registration Authority centrale**, eventualmente interoperante con una Struttura Anagrafica dell'Azienda, a supporto delle Certification Authority semplici interne al Sistema Informativo dell'Azienda;
2. una o piú **Certification Authorithy centrali** di tipo **semplice**, interoperanti con la Registration Authority centrale, ed eventualmente con la Struttura Anagrafica;
3. i moduli di gestione delle **attività di richiesta, acquisizione e gestione** locale dei **certificati X.509 degli utenti** operanti all'interno di un gruppo di utenza interno all'Azienda;
4. i moduli di gestione delle **attività di firma dei documenti e di verifica dei documenti firmati**, con l'utilizzo di certificati X.509 semplici, qualificati o accreditati, sia su dispositivi hardware (SmartCard, eToken) che su dispositivi software.

4Sign comprende:

- un Database centrale di Registration e Certification Authority (RA/CA-DB) per la RA e le CA interne
- un Web di amministrazione di RA/CA-DB per la RA e le CA interne
- un insieme di CGI per la gestione on-line del Client di Gestione dei Certificati per la RA e le CA interne
- un Client di Gestione dei Certificati per la RA e le CA interne
- un insieme di moduli Client di Firma Digitale (con certificate X.509 sia semplici che accreditati)
- un insieme di Librerie Dinamiche (server e client) per la gestione crittografica dei certificati.

Il database centrale RA/CA-DB (utilizzato sia per le operazioni di amministrazione centrali che per il supporto delle operazioni degli utenti per la richiesta e l'emissione di certificati) è organizzato in più sezioni:

- Tabelle Operatori (operatori, profili, log)
- Tabelle di RA (licenze, ruoli, utenti)
- Tabelle di Supporto RA (nazioni, regioni, province, comuni)
- Tabelle di CA (CA, policy, ruoli, richieste, certificati, CRL)
- Tabelle degli Eventi (eventi, codici)

4Sign è una infrastruttura **multi-CA**, in grado di gestire in outsourcing strutture RA/CA di firma per conto terzi, o comunque di differenziare le *line* applicative interne ad una azienda (commerciali, finanziarie, produttive, …) che richiedono strutture di RA/CA personalizzate.

Ogni entità esterna gestita in outsourcing e/o ogni *line* applicativa interna viene identificata da 4Sign mediante una **Licenza**, ovvero di un **Gruppo Chiuso di Utenza**. All'interno di ogni Gruppo 4Sign gestisce una o più CA. 4Sign è quindi orientato alla **certificazione degli utenti nell'ambito di Gruppi Chiusi di Utenza** (Licenze) per la **gestione della firma digitale** dei documenti.

La **Registration Authority di 4Sign è unica**, in quanto i compiti di notarizzazione affidati ad una RA sono indipendenti dalle CA gestite e dai relativi gruppi. Per contro 4Sign gestisce **differenti CA per la singola RA** nell'ambito di un unico database centrale, identificando le singole CA in base ad un codice identificativo univoco. 4Sign non certifica gli utenti finali se non sono associati ad un gruppo chiuso di utenza (licenza).

La **Registration Authority di 4Sign** può essere **facilmente integrata con la Struttura Anagrafica dell'Azienda**, limitando la sola visione anagrafica di 4Sign ai dati minimi necessari a svolgere le funzioni di riconoscimento certo degli utenti e ad alimentare le Certification Authority interne.

I moduli di gestione delle **attività di firma dei documenti e di verifica dei documenti firmati** possono essere **adottati singolarmente** per integrare le funzioni di **Firma Digitale nel contesto applicativo del Sistema Informativo dell'Azienda**, utilizzando certificati X.509 sia emessi internamente dall'Azienda, sia acquisiti da enti certificatori esterni. Nel panorama Italiano i moduli client di 4Sign sono interoperabili con i certificati emessi da Infocamere, da Actalis e da PosteCert: comunque l'approccio open-source e l'adozione di componenti che si basano sullo standard PKCS#11 per la gestione dei dispositivi sicuri di firma (SmartCard, eToken) garantisce virtualmente l'adozione di qualunque certificato X.509 aderente alle normative Europee e Italiane.

Inoltre 4Sign comprende un modulo separato, **4Ever**, orientatato a **gestire la firma digitale di stream**, ovvero di **sequenze di messaggi di log o di frame grafiche e/o sonore**, in modalità non presidiata, mantenendo il pieno valore di sicurezza e legale della firma apposta.

Architettura Generale e Componenti

L'implementazione della firma digitale nell'ambito di un Sistema Informativo Aziendale viene effettuata mediante la realizzazione, l'attivazione e la gestione dei seguenti componenti software:

- **Database Centrale di Registration e Certification Authority (RA/CA-DB)**
  Database consultabile mediante SQL, accessibile via ODBC o in modalità embedded, nel quale sono mantenute e gestite tutte le informazioni necessarie alla registrazione notarile degli utenti (RA), la registrazione dei gruppi di utenza (licenze), le CA collegate ai gruppi di utenza, le richieste ricevute dagli utenti, i certificati emessi, le liste di revoca delle CA.

- **Database del Giornale di Controllo delle Certification Authority (CAJG-DB)**
  Database consultabile mediante SQL, accessibile via ODBC o in modalità embedded, nel quale sono registrati gli eventi significativi della vita di ogni CA (le richieste ricevute dagli utenti, i certificati emessi, le liste di revoca delle CA).

- **Database del Giornale di Controllo della Time Stamp Authority (TSAJG-DB)**
  Database consultabile mediante SQL, accessibile via ODBC o in modalità embedded, nel quale sono registrati tutti gli eventi significativi della vita della TSA (marche temporali rilasciate).

- **Web di Amministrazione di RA/CA**
  Sito Web per l'amministrazione del database di Registration e Certification Authority: accesso consentito solo tramite certificati X.509 di autorizzazione, esclusivamente da parte di utenti che operano all'interno della rete protetta (Intranet) del Sistema Informativo Aziendale.

- **Web di Firma Forte delle CA**
  Sito Web per la gestione delle CA che operano con firma forte (dispositivo hardware quale SmartCard o eToken): permette ai responsabili delle CA di creare le nuove CA, di firmare le richieste di certificazione (rilasciando certificati utente firmati dalla CA), di firmare le CRL rilasciate dalla CA.

- **Server NTP (Network Time Protocol)**
  Di tipo stratum-1 o stratum-2 per l'erogazione di data/ora certa alla rete del Sistema Informativo Aziendale: questo server deve essere interconnesso con server pubblici (Internet) di tipo stratum-1 e stratum-2, per garantire il miglior allineamento della propria data/ora alla data/ora di Internet, anche a fronte della non disponibilità di uno o più server esterni.

- **Server TSA (Time Stamp Authority)**
  Sito Web con funzionalità TSA per il rilascio di marche temporali certificate: opera con un certificato di TSA rilasciato da una CA interna, a ciò dedicata. È aperto ai gruppi chiusi di utenza che devono effettuare le operazione di firma dei documenti con marcatura temporale.

- **Server OCSP (Online Certificate Status Protocol)**
  Server di rete per la verifica di validità dei certificati utente. Viene attivato con i dati di una CRL, ed alimentato con la stessa cadenza proprio delle CRL di una CA: è aperto al pubblico (gruppi chiusi di utenza) per la verifica di validità certificati.

- **Server LDAP (Lightweight Directory Access Protocol)**
  Database di rete per la pubblicazione dei certificati emessi e delle CRL (Certificate Revocation List) delle CA. Viene alimentato costantemente con i certificati emessi dalle singole CA e con le relative CRL: è aperto al pubblico (gruppi chiusi di utenza) per la verifica dei certificati e delle CRL.

- **Server 4Ever di Log Sicuro Firmato**
  Server 4Ever+4Elog che permette di raccogliere messaggi di log generati da programmi applicativi distribuiti sulla rete, e di firmarli digitalmente con le credenziali di firma di un operatore certificato prima di registrarli su un file centrale di log (opera con tecniche di protezione Stealth-RAM delle credenziali di firma operatore).

- **Client di Gestione dei Certificati Utente**
  Applicativo Client di Gestione dei Certificati utente: richiesta e scarico dei certificati, gestione locale dei certificati sulle workstation degli utenti.

- **Supporto on-line ai Client di Gestione dei Certificati Utente**
  Sito Web dedicato e protetto dagli accessi esterni, su cui opera un insieme di CGI per la gestione on-line dei Client di Gestione dei Certificati. Le CGI accedono al database RA/CA-DB per il login degli utenti il supporto alla preparazione delle richieste, l'acquisizione delle richieste di certificazione, il rilascio dei certificati emessi.

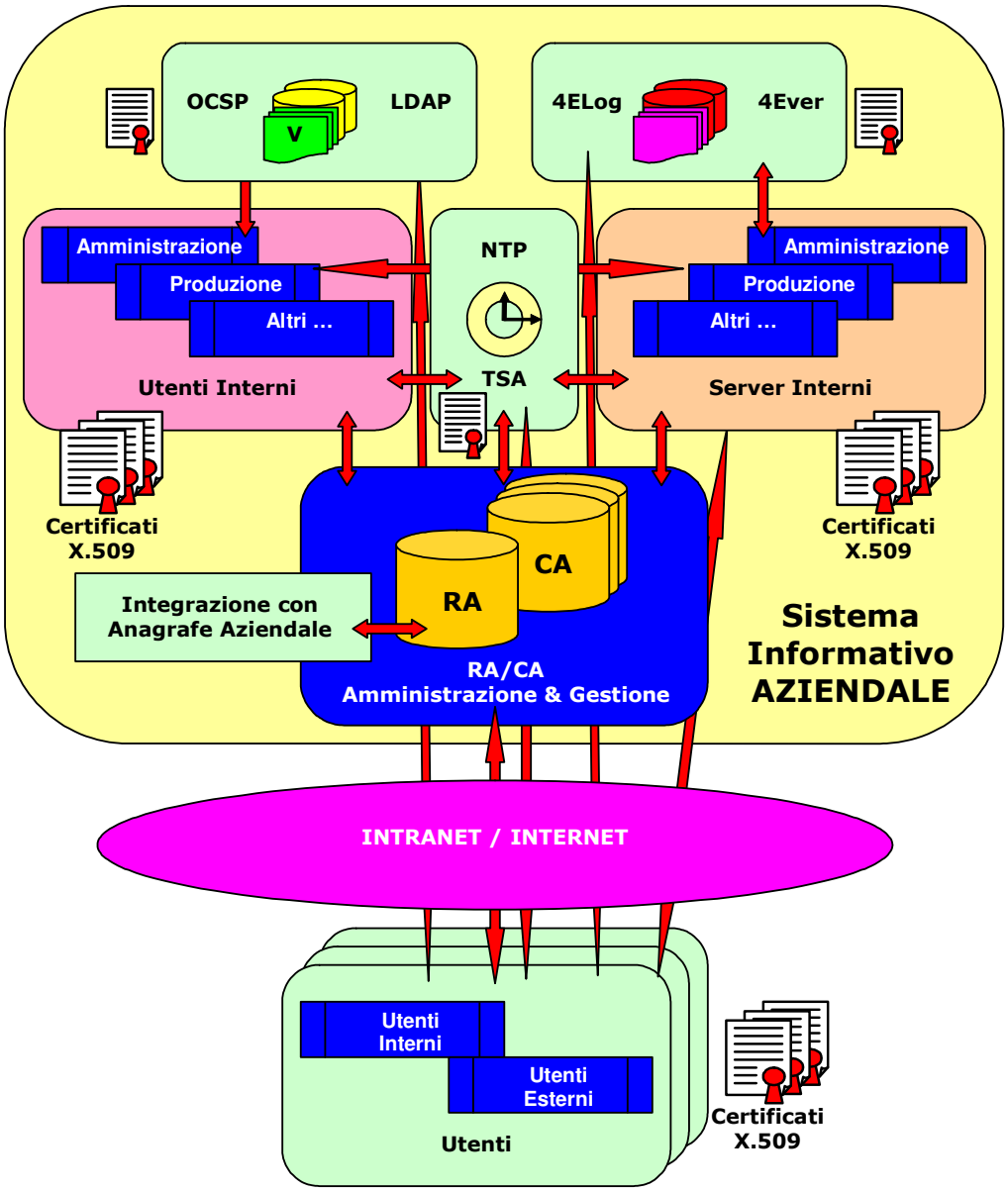
- **Client di Firma Digitale dei Documenti**
  Applicativo Client di Firma Digitale, in grado di richiedere marche temporali certificate al server TSA, per la firma digitale dei documenti utente, la produzione di documenti firmati con data certa, la verifica di documenti firmati con accesso alle liste di revoca delle CA emittenti.

- **Componenti di Crittografia**
  Librerie statiche e dinamiche che implementano le API necessarie alla gestione crittografica dei certificati, delle richieste, delle liste di revoca, eccetera. Organizzate in due gruppi specializzati, uno per i programmi server e unp per i programmi client.

- **Componenti di Verifica dei Documenti Firmati**
  Libreria utilizzabile sia sui server applicativi centrali che dal programma applicativo Client di Firma Digitale per effettuare la verifica di firma dei documenti, della data certa e delle liste di revoca della CA emittenti.

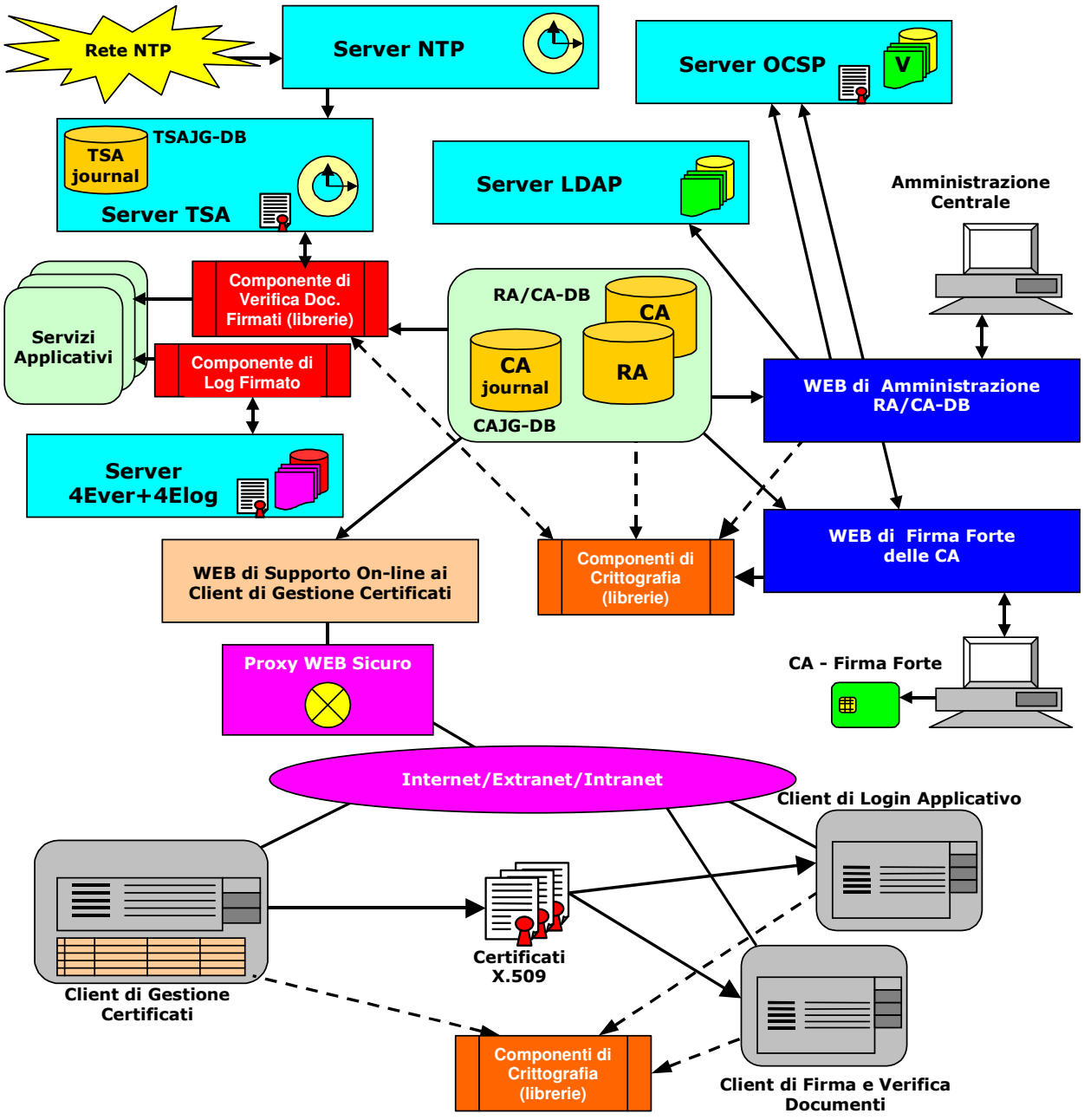
La figura 1 a pagina seguente seguente illustra l'architettura generale, evidenziando, all'interno del Sistema Informativo Aziendale:

- l'**infrastruttura centrale** di Amministrazione e Gestione della RA e delle CA;
- il **sottosistema di gestione temporale** e delle marche temporali NTP-TSA;
- il **sottosistema di pubblicazione e verifica di certificati e CRL** (LDAP e OCSP);
- il **sottosistema sicuro di raccolta e firma dei log** applicativi (4Ever+4Elog);
- l'**insieme dei server interni del Sistema Informativo Aziendale** che possono effettuare la firma automatica di documenti, azioni, messaggi per conto degli operatori del sistema;
- l'**insieme degli utenti interni**, che utilizzano il sistema di firma digitale per la richiesta di certificati di navigazione e di firma e per la firma di documenti;
- l'**insieme degli utenti esterni**, che utilizzano il sistema di firma digitale per la richiesta di certificati di navigazione e di firma e per la firma di documenti.



Architettura Generale di 4Sign

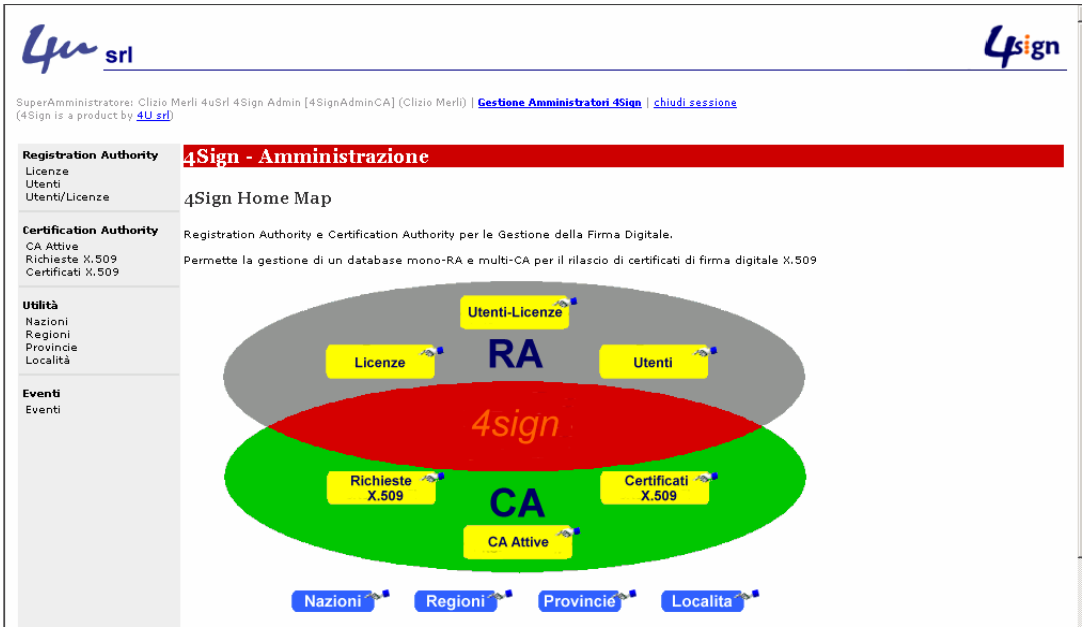
La figura seguente illustra l'architettura in termini dei componenti.



Componenti Software 4Sign per la Firma Digitale

Amministrazione RA/CA e Gestione dei Certificati

Il sito **Web di Amministrazione di RA/CA** è accessibile solo a livello centrale in modalità protetta (https), con il supporto di certificati di navigazione sia server che client. I certificati di navigazione client sono utilizzati dal sito per autenticare e autorizzare gli operatori centrali (amministratori), in base alle relative profilature funzionali.



4Sign: Modulo WEB di Amministrazione RA/CA

L'interazione tra l'amministrazione centrale RA/CA e gli utenti che richiedono la certificazione viene effettuata, lato utente, mediante un apposito modulo, **Client di Gestione dei Certificati**, che permette di:

- preparare le richieste dei certificati per gli utenti associati a un gruppo di utenza interno, con il supporto del RA/CA-DB tramite le User-CGI, e l'adozione di dispositivi crittografici locali, sia software che hardware;
- trasmettere le richieste di certificazione al centro per la firma;
- ricevere dal centro i certificati X.509 firmati da una CA;
- registrare localmente i certificati X.509 firmati (unitamente alle rispettive chiavi segrete e ai certificati della CA firmataria) nei dispositivi crittografici locali;
- esportare i certificati X.509 di firma debole in formato PKCS12.

Programmi Client di Firma e Verifica

Per le operazioni di firma e verifica l'utente può utilizzare:

- un Client di Firma e Verifica dei Documenti
- un Client di Firma dei Login

Il Client di Firma e Verifica dei Documenti è una applicazione che permette di:

- selezionare e visionare i documenti da firmare;
- effettuare la firma digitale del documento selezionato con il certificato selezionato, generando un file di tipo PKCS7 contenente il documento e la firma;
- effettuare la marcatura temporale (opzionale) del documento firmato, con una marca certificata ottenuta on-line dal server TSA centrale (con il supporto delle CGI utente del WEB di Supporto On-line), e generare un file di tipo S/MIME contenente il file PKCS7 (documento firmato) e la marca temporale certificata;
- effettuare la verifica della validità della marca temporale (opzionale) apposta su un documento firmato in formato S/MIME (con il supporto delle CGI utente del WEB di Supporto On-line);
- verificare la validità formale di un documento firmato (in formato PKCS7), estraendo il documento originale dopo la verifica;
- verificare la validità del certificato utilizzato per la firma al momento della firma, con il supporto delle CGI utente del WEB di Supporto On-line (opzionale);
- visualizzare un documento verificato.

Il Client di Firma e Verifica di Login è una applicazione che permette di:

- generare un SID (Session Identifier) univoco per una sessione, e di salvarlo su un file;
- effettuare la firma digitale del SID, generando un file di tipo PKCS7 contenente il SID e la firma;
- effettuare la marcatura temporale (opzionale) del SID firmato, con una marca certificata ottenuta on-line dal server TSA centrale (con il supporto delle CGI utente del WEB di Supporto On-line), e generare un file di tipo S/MIME contenente il file PKCS7 (SID firmato) e la marca temporale certificata.

Utilizzo dei Certificati: Firma di un Documento

La firma digitale di un documento viene effettuata con l'Applicativo Client di Firma Digitale, che utilizza i certificati di firma emessi dalle CA interne o esterne. Il processo di firma prevede:

1. la selezione e visualizzazione del documento da firmare (qualunque file accessibile dalla work-station del firmatario);
2. selezione del dispositivo sicuro di firma da utilizzare, e del certificato X.509 tra quelli contenuti a bordo del dispositivo selezionato;
3. digitazione del PIN di protezione del certificato, e visualizzazione degli elementi di identificazione;
4. firma digitale del documento, con generazione contestuale di un file PKCS7 (di tipo .p7m) contenente il documento originale, la sua impronta firmata, e il certificato di firma dell'utente;
5. opzionalmente calcolo della impronta del file PKCS7 generato, e contestuale richiesta di firma a un server TSA (Time Stamp Authority), che ritorna l'impronta e la marca temporale (data/ora certa) firmate con il proprio certificato di TSA (marca temporale firmata).

Verifica di Firma di un Documento

La verifica di firma digitale apposta su un documento può essere effettuata da chiunque (utenti, server, operatori, amministratori) abbia la necessità di verificare la validità di un documento firmato. La verifica prevede:

- Per documenti S/MIME di tipo .m7m:
  1. estrazione del file PKCS7 (.p7m) e della relativa marca temporale;
  2. verifica formale della marca temporale;
  3. verifica di validità del certificato di TSA associato;
  4. verifica del file PKCS7 (.p7m) secondo i passi seguenti.

- Per documenti PKCS7 di tipo .p7m:
  1. estrazione del documento originale, dell'impronta firmata, e del certificato di firma
  2. ricalcolo dell'impronta del documento originale
  3. decodifica dell'impronta firmata con la chiave pubblica del certificato e verifica con l'impronta ricalcolata;
  4. verifica di validità del certificato di firma.

Utilizzo di un Certificato: Firma di Login Applicativo

L'accesso via WEB a funzioni applicative particolarmente sensibili del Sistema Informativo aziendale può essere controllato mediante l'adozione un certificato X.509 client, mediante il quale non sono controllate solamente l'autenticazione e l'autorizzazione, ma viene firmata l'azione iniziale di login (marca di login).

La marca di login è un file che riporta, tra l'altro, un identificato univoco di sessione, che viene riproposto dal browser utente ad ogni azione successiva al login, e che può essere utilizzato dalle CGI applicative interessate per effettuare il log transazionale sicuro sui database applicativi.

Nell'ambito di una sessione la prima azione ad essere riportata sul log è il login con associata la marca di login firmata digitalmente dall'utente, e contenente un token di accesso univoco. Tutte la azioni successive riportate sul log, in quanto riportano lo stesso token di accesso firmato inizialmente dall'utente, sono implicitamente firmate dall'utente: la garanzia di ciò viene data da un processo di firma automatica, che opera a bordo del server di log, e che viene attivato in modo certificato, con certificato di firma X.509 di un operatore centrale. Quindi a certificare la sequenza delle azioni utente è il processo di firma automatica dei log, che firma in nome dell'operatore che lo ha attivato.

La firma di login viene effettuata con il supporto dell'Applicativo Client di Login, secondo i seguenti passi:

1. l'Applicativo Client di Login per la validazione iniziale dell'utente viene richiamato localmente al sistema client su cui opera l'utente da una componente applicativa di login (componente locale di un applicativo client/server, ActiveX, applet di login, eccetera): il richiamo viene effettuato passando come parametro l'identificatore dell'utente;
2. l'Applicativo Client di Login genera un identificatore univoco di sessione (SID – Session Identifier) che viene scaricato su un file di lavoro locale: il SID deve contenere tra l'altro l'identificativo dell'utente, l'indirizzo IP del sistema client, la data/ora di login;
3. l'Applicativo Client di Login propone il SID per la firma all'utente;
4. l'utente firma digitalmente il file contenente il SID (in modo analogo a quanto dettagliato al paragrafo 8.3.1) completando eventualmente la firma con l'apposizione di una marca temporale certificata dalla TSA: il SID firmato (Token di Accesso) viene salvato su un file locale al sistema client;
5. la componente applicativa di login effettua il log dell'azione firmata di login (e quindi del SID e del Token di Accesso).

Tutte le azioni intraprese successivamente dall'utente nell'ambito della sessione applicativa mantengono il SID e il relativo Token di Accesso, creando sui log centrali una catena firmata di azioni (e quindi di transazioni). Si veda al riguardo il componente di sicurezza 4Ever: la catena di azioni di login associate ad una singola sessione utente viene firmata elettronicamente a livello centrale con l'ausilio del componente di sicurezza 4Ever.

Verifica di Firma di Login Applicativo

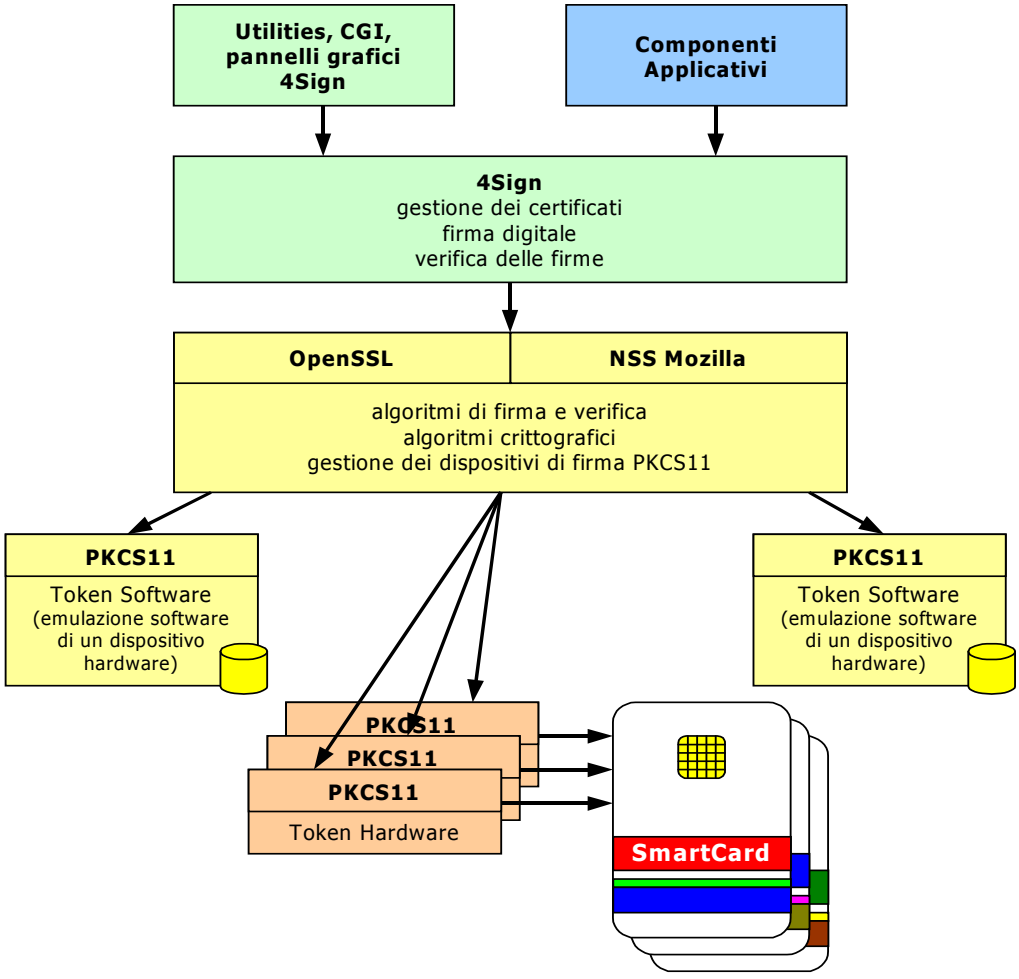
L'azione di verifica di un login applicativo, e della conseguente catena di azioni/transazioni, viene effettuata centralmente secondo i seguenti passi:

1. estrazione da un file di log firmato della catena di log, che inizia con il login (SID) interessato;
2. verifica della firma digitale apposta sui file di log da cui sono stati estratti gli elementi della catena;
3. verifica della firma digitale apposta dall'utente sul primo elemento della catena (azione di login).

Librerie 4Sign

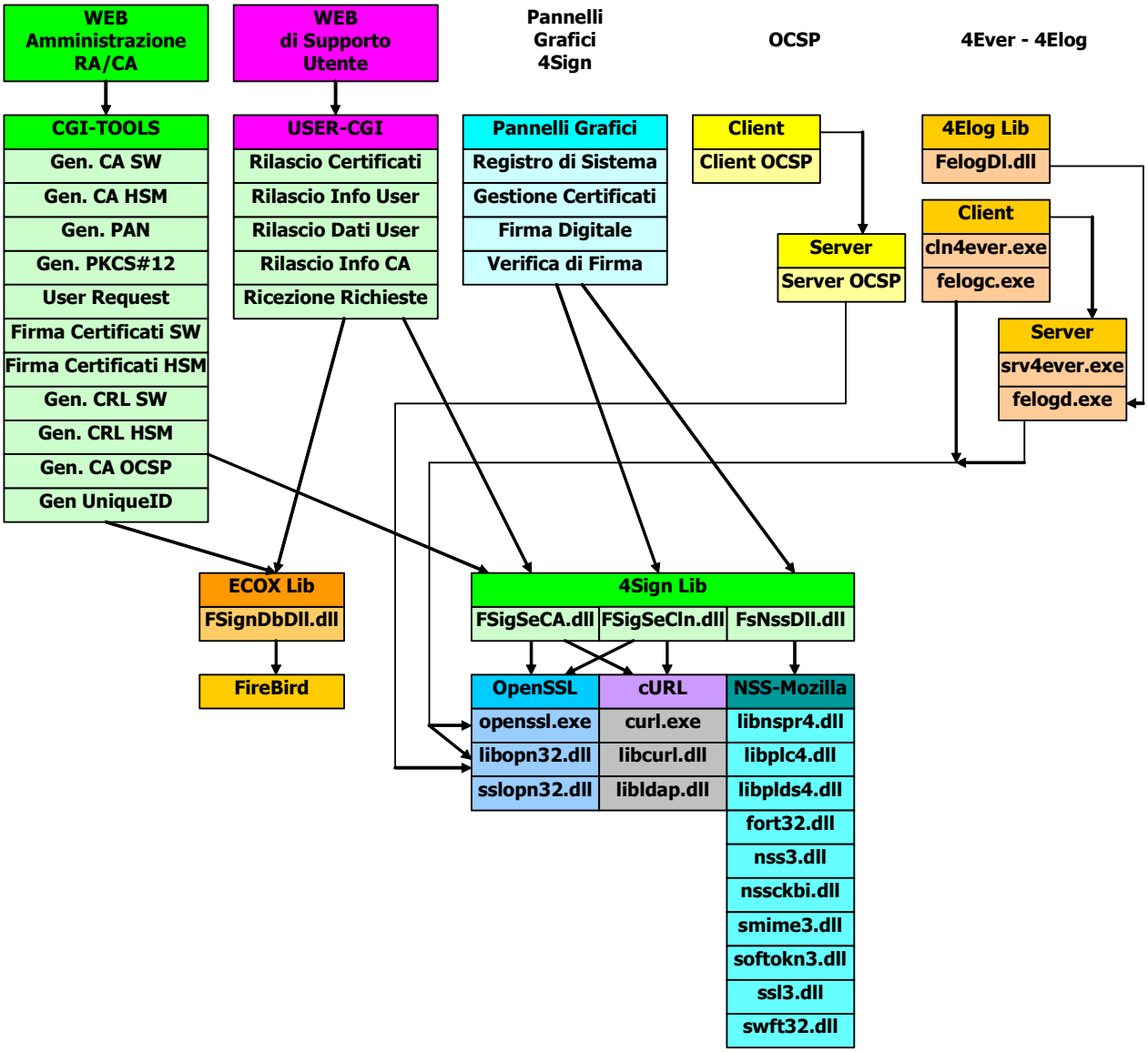
La suite 4Sign comprende, quale elemento centrale, un insieme di librerie statiche e dinamiche di interfaccia verso i componenti opensource di crittografia. Lo schema seguente illustra la struttura funzionale di 4Sign, comprendente:

- un livello applicativo: utilities, CGI e pannelli grafici 4Sign; componenti applicativi utente;
- un livello 4Sign di alto livello, per la gestione dei certificati, la firma digitale, la verifica delle firme, eccetera;
- un livello opensource intermedio, per gli algoritmi di firma e verifica, gli algoritmi crittografici, la gestione dei dispositivi di firma su interfaccia PKCS11;
- un livello proprietario di basso livello, costituito dalle librerie dinamiche PKCS11 fornite dai costruttori dei dispositivi sicuri, dai driver di accesso ai lettori dei dispositivi.



Struttura Funzionale 4Sign

I componenti 4Sign sono illustrati nella figura seguente.



Componenti Software 4Sign®

4Ever®: Firma Digitale di Stream

4Ever® è un prodotto software della suite 4Sign orientatato a gestire la firma digitale di stream, ovvero di sequenze di messaggi di log o di frame grafiche e/o sonore, in modalità non presidiata, mantenendo il pieno valore legale della firma apposta. Lo scenario tipico di 4Ever® prevede:

- Uno o più processi applicativi (**Target Generator**), di generazione delle sequenze di eventi (messaggi, frame) da firmare (**Stream Target**);
- un processo di firma (**4Ever-Server**) attivato da un operatore con le proprie credenziali di firma, che operando in background verifica con continuità la generazione degli Stream Target e provvede alla firma digitale del relativo file di log utilizzando l'infrastruttura 4Sign;
- un processo collettore (**4Elog-Server**) attivato contestualmente a 4Ever-Server, che raccoglie le sequenze di dati da firmare dai processi Target-Generator, e li accumula nel file di log per essere firmati dal processo Server 4Ever;
- un client di attivazione/disattivazione del servizio (**4Ever-Client**), utilizzato degli operatori per attivare/disattivare il processo di firma digitale dei Target.

Il processo 4Ever-Server mantiene, durante la sua vita attiva, le credenziali di firma dell'operatore in un'area di memoria protetta da uno schema proprietario (**MutingBox**), in base al quale il contenuto protetto varia con continuità la sua forma senza perdere le proprie informazioni. Questo approccio permette di garantire la validità della firma digitale apposta con continuità dal 4Ever-Server ai file dello Stream Target senza la necessità di un continuo presidio dell'operatore.

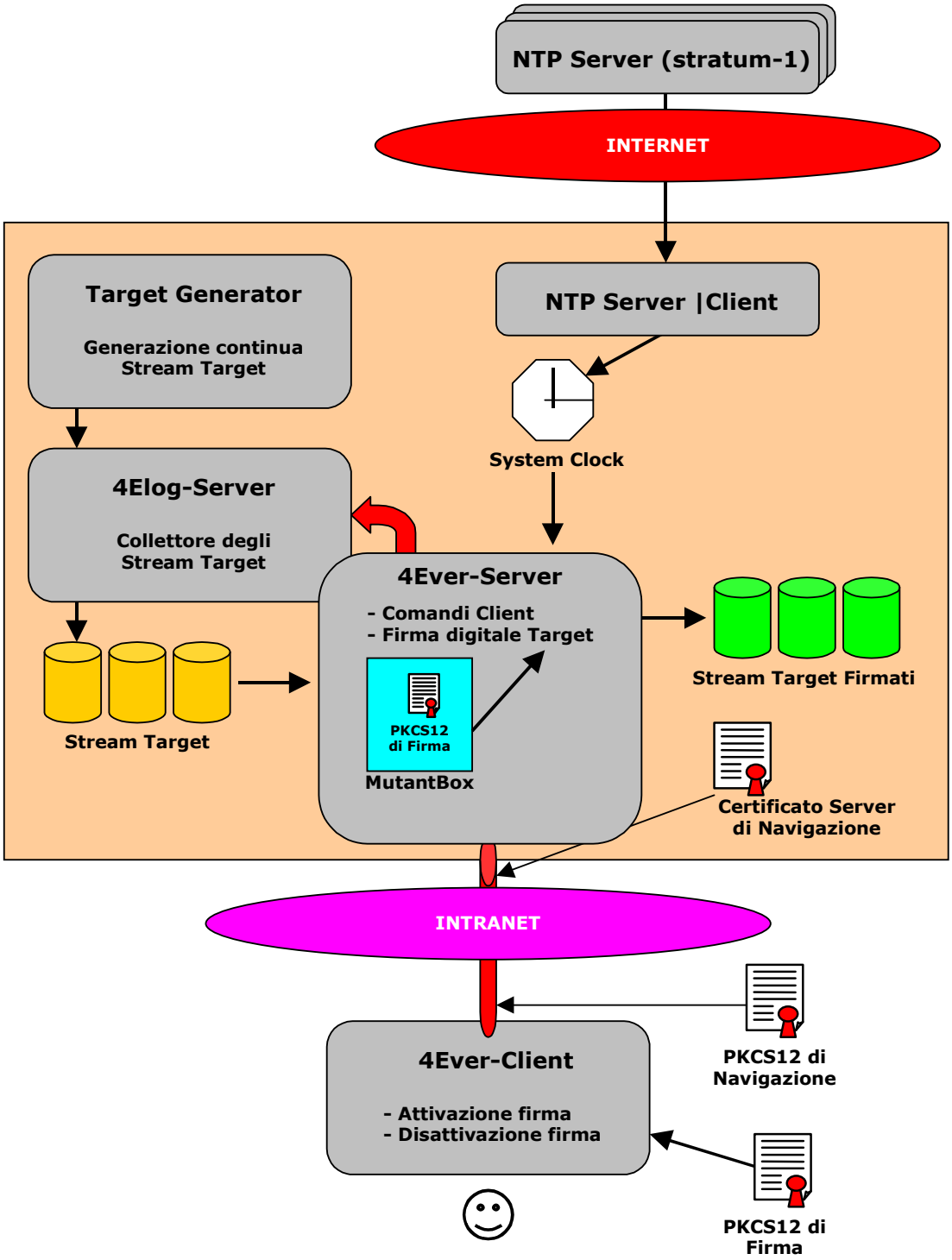
4Ever-Server puó limitare la propria azione alla generazione di **file firmati in formato PKCS7**, o estendere il processo di firma alla **apposizione di una marca temporale auto-certificata**, generando **file SMIME** contenenti sia i file PKCS7 firmati che la marca temporale di tali file.

A tal scopo è necessario che sul sistema su cui opera il servizio 4Ever® opera sia attivo un processo di mantenimento della data/ora di sistema in base al protocollo NTP, e che tale processo sia costantemente attivo e connesso tramite la rete Internet ai propri nodi NTP di primo livello (stratum-1) per garantire l'allineamento della data/ora di sistema.

Per ogni Stream Target da firmare può essere attivato un servizio 4Ever® dedicato, che risponde al proprio 4Ever-Client su una porta TCP riservata. Su un sistema possono operare piú coppie 4Ever-Server/4Elog-Server per la firma di differenti Stream Target: ogni istanza del servizio utilizza una porta TCP riservata per ricevere i comandi client, e una porta TCP riservata per ricevere i dati da accumulare nel file di log.

La comunicazione tra 4Ever-Client e 4Ever-Server viene effettuata mediante una connessione SSL/TCP, con l'adozione di appositi certficati di navigazione. In fase di attivazione 4Ever-Client trasmette a 4Ever-Server, su questa connessione certificata, le credenziali di firma in formato PKCS12, opportunamente protette da una frase di passo che viene trasmessa in formato protetto. La frase di passo viene riassegnata automaticamente dal 4Ever-Server. Le credenziali dell'operatore e la relativa frase di passo sono mantenute dal 4Ever-Server nell'area di memoria mutante (MutantBox).

L'architettura 4Ever è illustrata nella figura seguente.



Architettura 4Ever

Processi e File

Il servizio **4Ever** è implementato con il supporto di due processi:

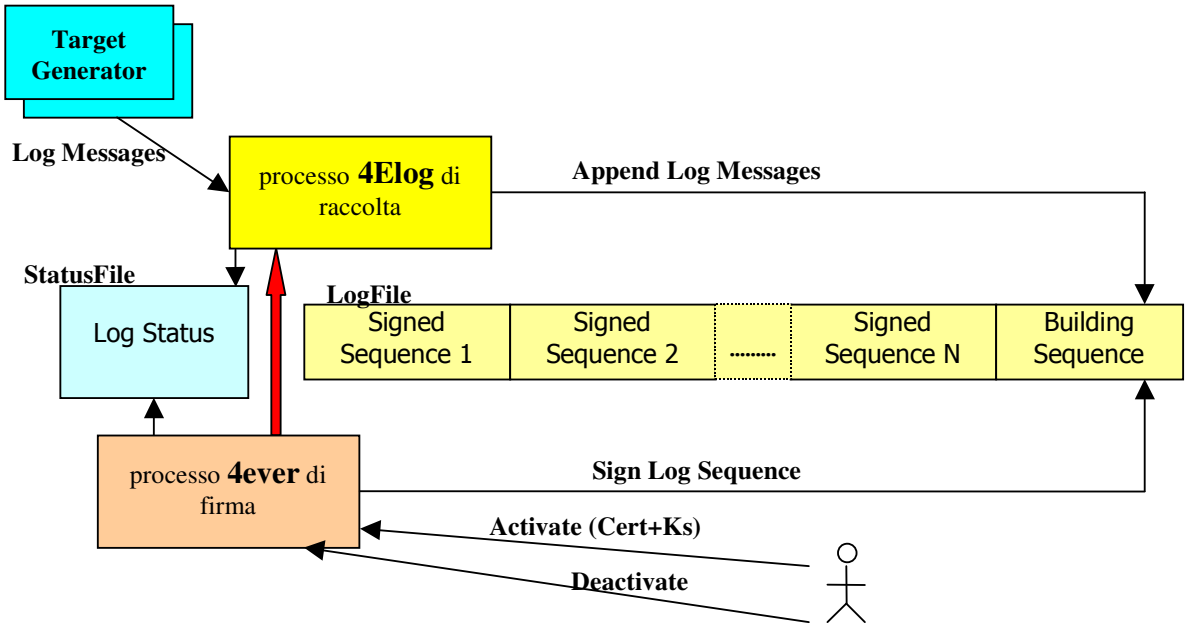
4Ever-Server processo principale, monothread-monoconnessione, di firma digitale dei log
4Elog-Server processo secondario, multithread, di raccolta dei log

I due processi operano con il supporto di due file:

StatusFile file contenente le variabili comuni di log, utilizzato per il sincronismo tra 4Ever e 4Elog
LogFile file su cui sono memorizzate le sequenze di log firmate

I log sono raccolti in un file di log (LogFile) come sequenze di messaggi (testata + dati), inizialmente non ancora firmati. Periodicamente il servizio 4Ever provvede a firmare l'ultima sequenza raccolta. Di conseguenza il LogFile è un insieme ordinato temporalmente di sequenze firmate ed eventualmente di una sequenza (l'ultima) non ancora firmata. Ogni sequenza è costituita da un insieme variabile di messaggi, in cui ogni messaggio viene marcato con l'Hash del messaggio precedente della sequenza (ad esclusione del primo che mantiene l'Hash della propria area dati).

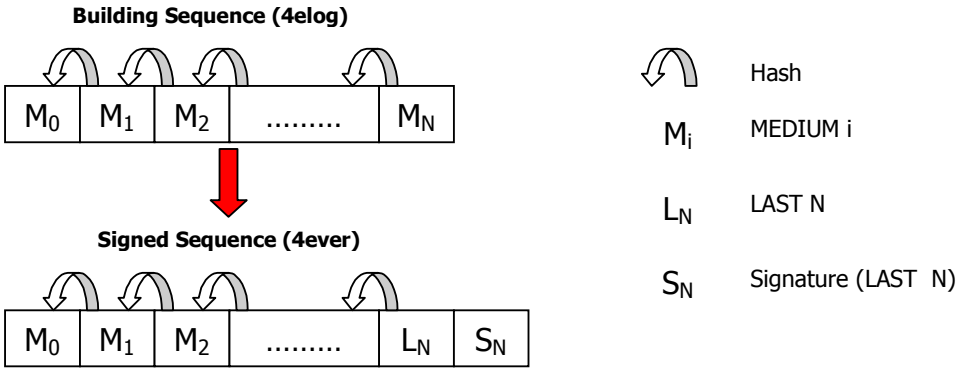
La figura seguente illustra lo schema del servizio 4Ever. Il processo 4Ever-Server viene armato per la firma da un operatore, che trasmette su un canale sicuro (SSL) il certificato e la chiave segreta di firma, che sono poi mantenuti in una meroria interna protetta (stealth memory di tipo MutantBox).



Quando viene armato il processo 4Ever-Server crea il processo 4Elog-Server, il quale opera su una porta TCP dedicata. I processi applicativi comunicano i messaggi di log al processo 4Elog-Server, che li accumula nel LogFile nell'ultima sequenza non firmata (Building Sequence). Il processo 4Ever-Server si attiva periodicamente, e firma l'ultimo messaggio della Building Sequence trasformandola in una Signed Sequence.

Validità della Firma di una Sequenza

La validità del meccanismo di firma su una Signed Sequence è dato dalla catena di Hash che ogni messaggio mantiene relativamente al messaggio precedente (ad eccezione del primo che mantiene l'Hash della propria area dati). La firma viene apposta sull'ultimo messaggio della sequenza. La figura seguente illustra il meccanismo di raccolta dei messaggi nella Building Sequence e l'azione di firma di quest'ultima che viene così trasformata in una Signed Sequence.



Architettura del Servizio 4Ever

La raccolta degli eventi viene effettuata da un insieme di **sensori distribuiti**, che sono operativi sulla rete del cliente finale, e da uno o più collettori periferici, **4Ever-Collector**. I sensori distribuiti possono interagire direttamente con il 4Elog-Server, o generare log in base allo standard 'syslog' indirizzati a uno o più processi 'syslogd' operanti sulla rete (collettori).

Un collettore è un server 'syslogd', su piattaforma Unix-like, che opera in accoppiamento con un plugin specifico, **4Ever-Plugin**: il demone 'syslogd' raccoglie gli eventi dai sensori distribuiti, e li passa al plugin che provvede a marcarli con l'identificativo del sensore (in base all'indirizzo IP) e a trasmetterli al processo 4Elog del 4Ever-Server per la registrazione. Nel caso di non raggiungibilità o non disponibilità del 4Ever-Server (interruzioni di rete o temporanea sospensione del servizio), il collettore 4Ever-Collector provvede a mantenere traccia degli eventi trasmessi, e a riattivare la trasmissione quando il 4Ever-Server è nuovamente raggiungibile/disponibile.

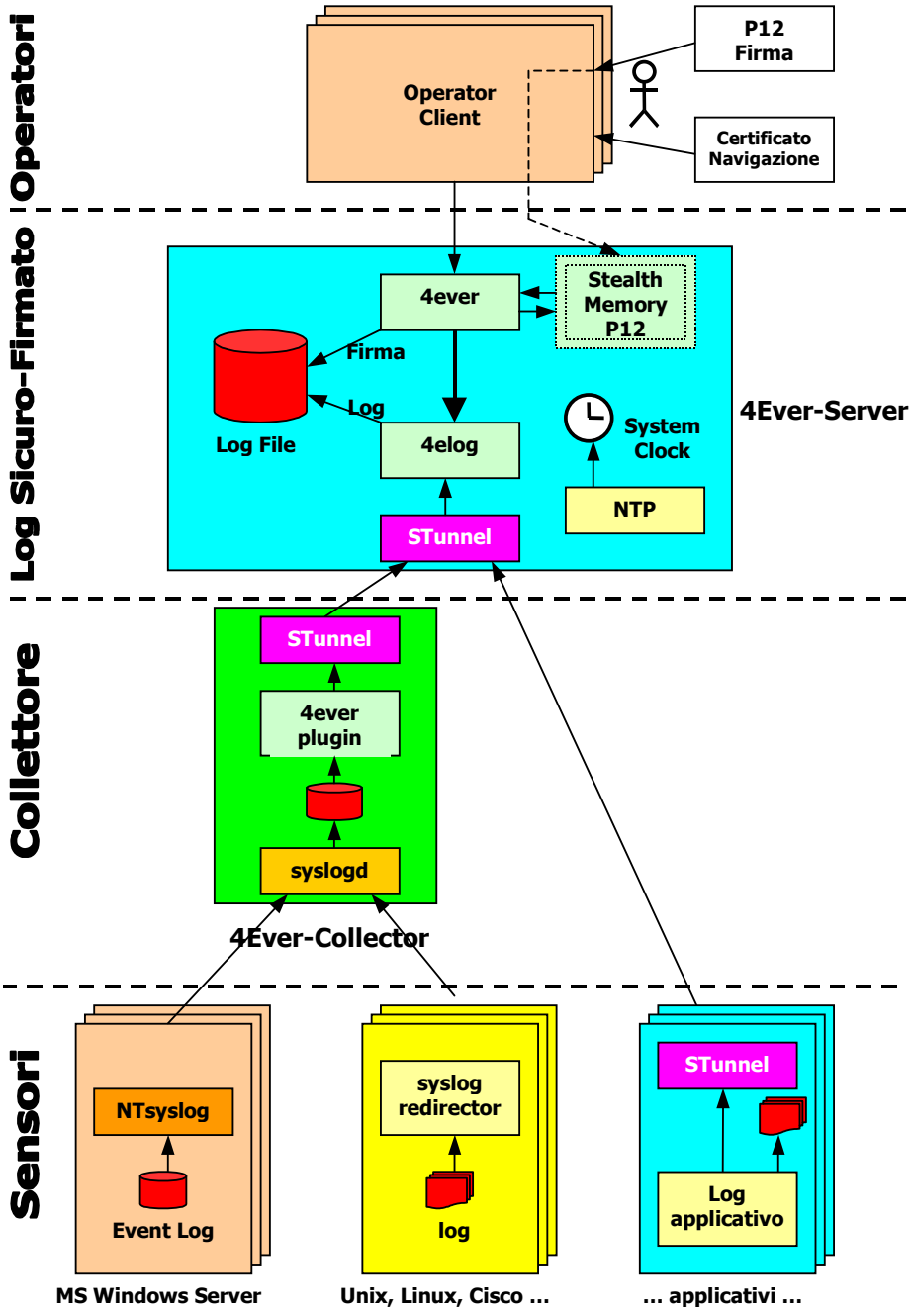
La comunicazione tra il plugin del 4Ever-Collector e il 4Ever-Server viene protetta utilizzando una coppia di processi **STunnel**, che realizzano un canale SSL gestito con i certificati X.509 propri del 4Ever-Collector e del 4Ever-Server.

Gli **operatori del servizio 4Ever** possono accedere da remoto al processo 4Ever del 4Ever-Server per attivare o disattivare il servizio, o per ricevere gli eventi relativi (firma delle sequenze, errori di firma, errori di sistema, rotazione del file di log).

L'accesso degli operatori viene effettuato mediante un protocollo dedicato trasportato su un canale SSL gestito con appositi certificati X.509 di navigazione client e un certificato X.509 server. Le operazioni di maintenance del 4Ever-Server possono essere effettuate solo dalla console locale di sistema.

La figura seguente illustra lo schema implementativo del sistema. In tale schema si identificano quattro livelli di rete:

1. i sensori applicativi e/o esterni, che sentono gli eventi e li inviano a un 4Ever-Collector;
2. il 4Ever-Collector che raccoglie gli eventi, li marca con l'identificazione di origine e li inoltra, mediante un canale sicuro, al 4Ever-Server;
3. il 4Ever-Server che riceve gli stream di eventi dai 4Ever-Collector, li registra localmente firmandoli digitalmente con il certificato X.509 e relativa chiave privata di firma del servizio;
4. gli operatori del 4Ever-Server che tramite un programma client possono, dalle rispettive postazioni, attivare, disattivare e monitorare il servizio 4Ever.



Approccio OpenSource

I componenti client (Client di Gestione dei Certificati e Client di Firma Digitale) e server (UserCGI e AdminCGI) di 4Sign sono stati sviluppati con l'obiettivo della piena e completa portabilità su piattaforme diverse da quella iniziale (Microsoft Windows). Di conseguenza sono stati adottati per lo sviluppo vari tool opensource, selezionati per essere interoperativi con differenti piattaforme (Unix, Microsoft Win32, in futuro MacOS). In particolare, per quanto riguarda Microsoft Windows, l'unico punto di riferimento è l'interfaccia Win32, comune a tutte le versioni di Windows.

I tool opensource adottati per lo sviluppo di 4Sign sono:

Apache Web Server opensource.

PHP package opensource che opera come Application Server associato ad Apache (per la realizzazione del sito di Amministrazione centrale, Admin-CGI).

FOX package opensource C++ di gestione dei pannelli grafici (opera in ambiente Unix, MacOs, Win32).

cURL package opensource di gestione della comunicazione client/server tra il Client di Gestione dei Certificati e le CGI (UserCGI) che operano al centro (RA/CA) sotto il controllo di un server WEB: opera come client HTTP (porta 80) o HTTPS (porta 443).

eCGI (easy CGI) package opensource di supporto alla implementazione delle UserCGI di accoglienza alle richieste del Client di Gestione dei Certificati: permette tra l'altro la facile gestione delle funzioni di upload, che potranno essere richieste nelle future versioni del Client di Firma Digitale per tasmettere al centro i file firmati.

OpenSSL package opensource per la gestione della crittografia e dei certificati X.509.

ModSSL package opensource per la gestione delle connessioni SSL/TSL con il Web Server Apache.

OpenTSA package opensource per la gestione delle TSA (Time Stamp Authority): è di fatto una estensione di OpenSSL e di Apache.

NTPS package opensource per la gestione di server NTP (Network Time Protocol) ed SNTP (Simple Network Time Protocol).

OpenLDAP package opensource per la gestione di server LDAP (Lightweigt Directory Access Protocol).

NSS-Mozilla package opensource C per la gestione della crittografia e dei certificati X.509 con interfaccia PKCS11 verso i dispositivi di gestione dei certificati (Token).

FireBird package opensource che implementa un DBMS (Data Base Management System) derivato e compatibile con InterBase di Borland.