



Lezione 6 – Trusted Execution Environment and Blockchain

Prof. Esposito Christian

Corso di Sicurezza dei Dati



::: Sommario

- Introduzione alle TEE;
- Applicazione nelle Blockchain:
 - Hyperledger Sawtooth e Avalon.

... Letture

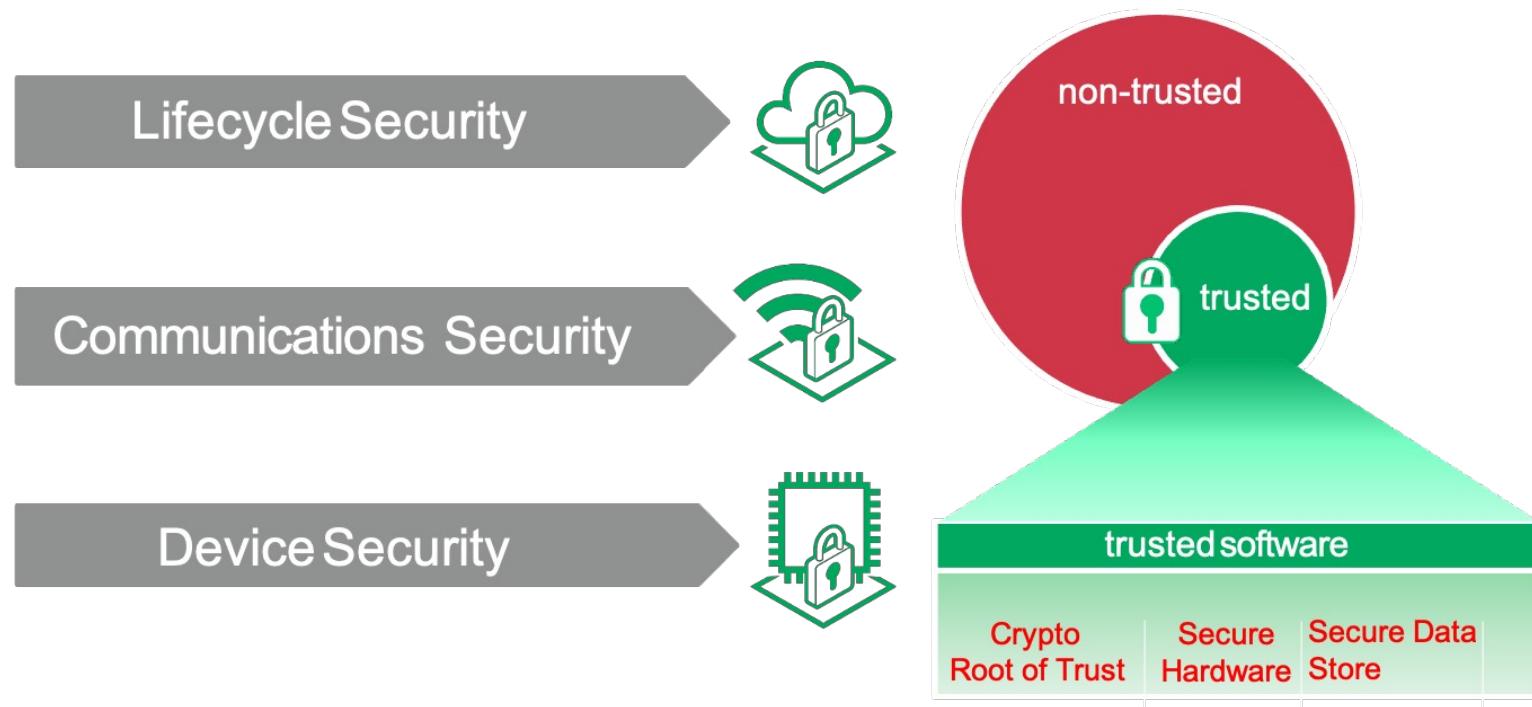
- M. Sabt, M. Achemlal, and A. Bouabdallah, "Trusted execution environment: what it is, and what it is not", Proceedings of the IEEE Trustcom/BigDataSE/ISPA Conference, 2015.
- Y. Wang, J. Li, S. Zhao and F. Yu, "Hybridchain: A Novel Architecture for Confidentiality-Preserving and Performant Permissioned Blockchain Using Trusted Execution Environment", in IEEE Access, vol. 8, pp. 190652-190662, 2020.
- K. Rilee, "Understanding Hyperledger Sawtooth—Proof of Elapsed Time," 21 02 2018. [Online]. Available: <https://medium.com/kokster/understanding-hyperledger-sawtooth-proof-of-elapsed-time-e0c303577ec1>. [Accessed 31 08 2018].
- Hyperledger Avalon: building the next wave of confidential applications, 3 10 2019 [Online] Available: <https://medium.com/iexec/hyperledger-avalon-building-the-next-wave-of-confidential-applications-54ba49dcd7e7>.



Introduzione a TEE

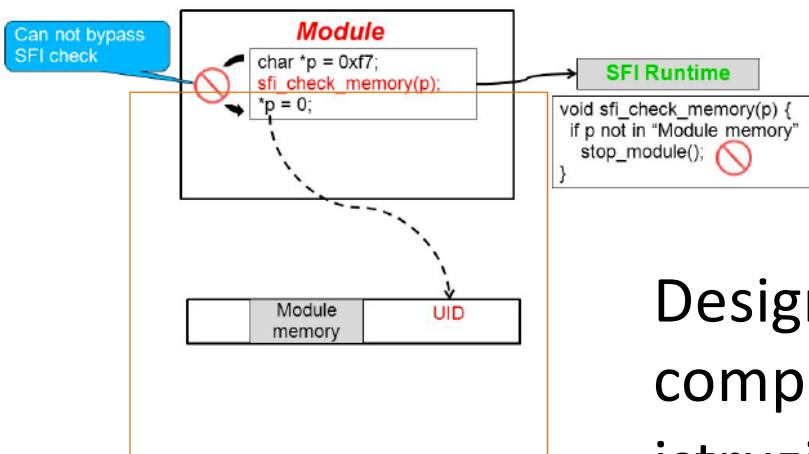
::: Introduzione

Il Trusted Computing (TC) ha lo scopo di garantire la sicurezza mediante la realizzazione di una sorta "cassaforte virtuale" attorno ai dati ed ai programmi. Se dati o programmi esterni vogliono avere accesso a questa cassaforte devono ottenere la chiave dal sistema di TC e solamente dati e programmi autentificati possono disporre delle risorse del sistema: dall'hard disk, alla memoria RAM, alla CPU.



::: Software Fault Isolation

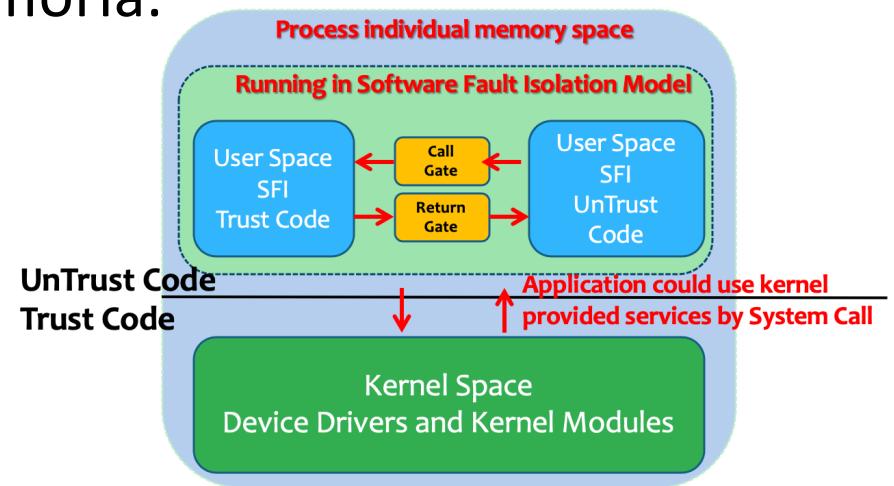
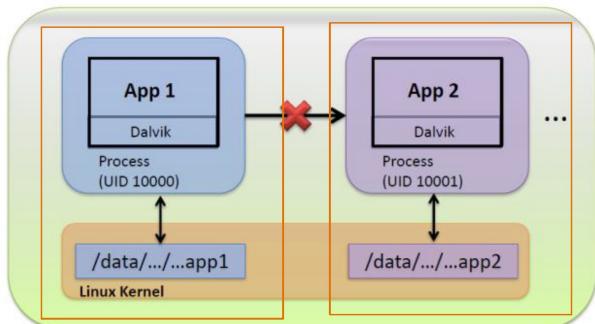
Software Fault Isolation (SFI[SOSP93])



SFI è una tecnica di strumentazione software a livello di codice macchina per stabilire domini di protezione logica all'interno di un processo.

Designa una regione di memoria per un componente non attendibile e fornisce istruzioni pericolose per limitare il suo accesso alla memoria.

È indicato come sandbox del codice.



::: Esecuzione Isolata (1/4)

Isolare l'esecuzione del codice è uno degli approcci fondamentali per ottenere la sicurezza. La virtualizzazione può creare un ambiente di esecuzione isolato per l'esecuzione di strumenti difensivi. Gli approcci esistenti basati sulla virtualizzazione hanno limitazioni, tra cui:

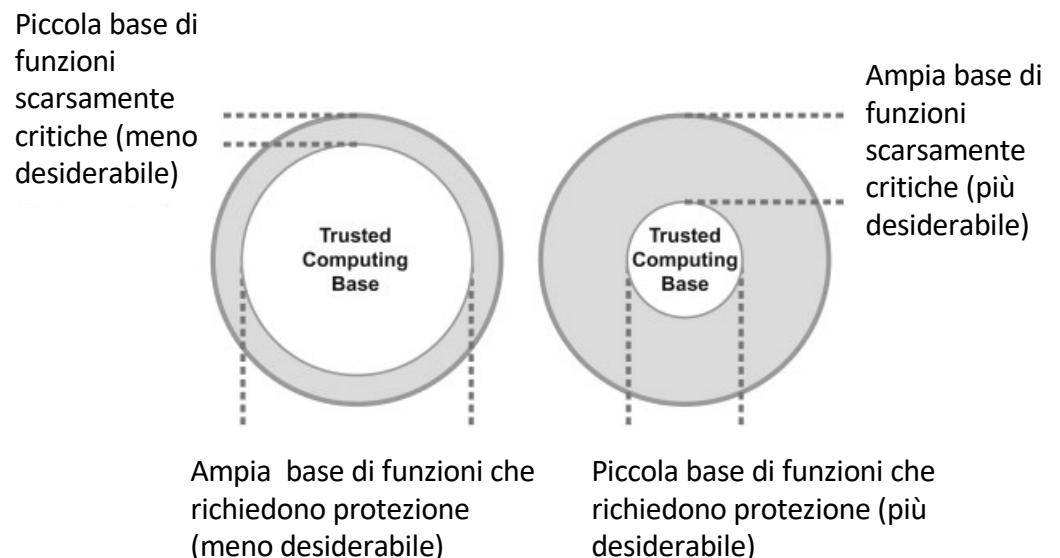
1. Dipendenza da hypervisor che possono avere un'ampia Trusted Computing Base (TCB).

::: Esecuzione Isolata (1/4)

Isolare l'esecuzione del codice è uno degli approcci fondamentali per ottenere la sicurezza. La virtualizzazione può creare un ambiente di esecuzione isolato per l'esecuzione di strumenti difensivi. Gli approcci esistenti basati sulla virtualizzazione hanno limitazioni, tra cui:

1. Dipendenza da hypervisor che possono avere un'ampia **Trusted Computing Base (TCB)**.

Il TCB di un sistema informatico è l'insieme di tutti i componenti hardware, firmware e/o software critici per la sua sicurezza: bug o vulnerabilità che si verificano all'interno del TCB potrebbero compromettere le proprietà di sicurezza dell'intero sistema.

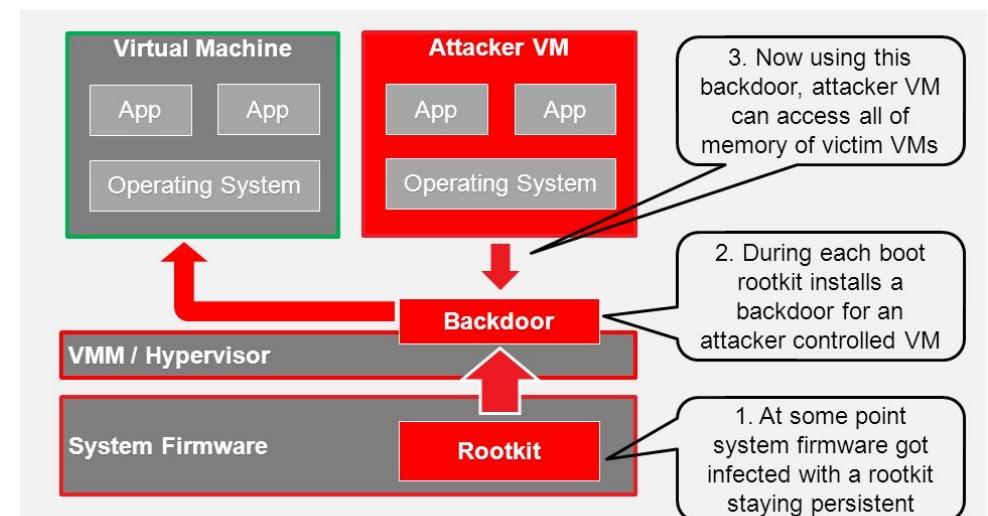


::: Esecuzione Isolata (1/4)

Isolare l'esecuzione del codice è uno degli approcci fondamentali per ottenere la sicurezza. La virtualizzazione può creare un ambiente di esecuzione isolato per l'esecuzione di strumenti difensivi. Gli approcci esistenti basati sulla virtualizzazione hanno limitazioni, tra cui:

1. Dipendenza da hypervisor che possono avere un'ampia Trusted Computing Base (TCB).
2. Mancata gestione dell'hypervisor o dei rootkit del firmware.

Un rootkit è una raccolta di software, in genere dannoso, progettato per consentire l'accesso a un computer o a un'area del suo software altrimenti non consentito.

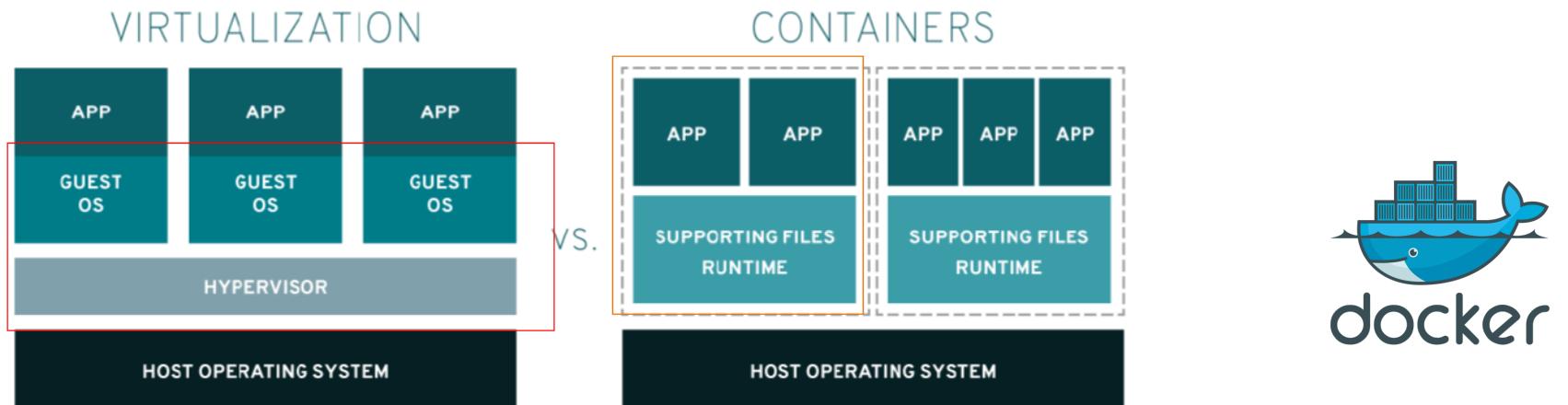


::: Esecuzione Isolata (1/4)

Isolare l'esecuzione del codice è uno degli approcci fondamentali per ottenere la sicurezza. La virtualizzazione può creare un ambiente di esecuzione isolato per l'esecuzione di strumenti difensivi. Gli approcci esistenti basati sulla virtualizzazione hanno limitazioni, tra cui:

1. Dipendenza da hypervisor che possono avere un'ampia Trusted Computing Base (TCB).
2. Mancata gestione dell'hypervisor o dei rootkit del firmware.
3. Soffre di sovraccarico delle prestazioni del sistema (ad esempio, cambi di contesto da una VM a un hypervisor).

::: Esecuzione Isolata (2/4)

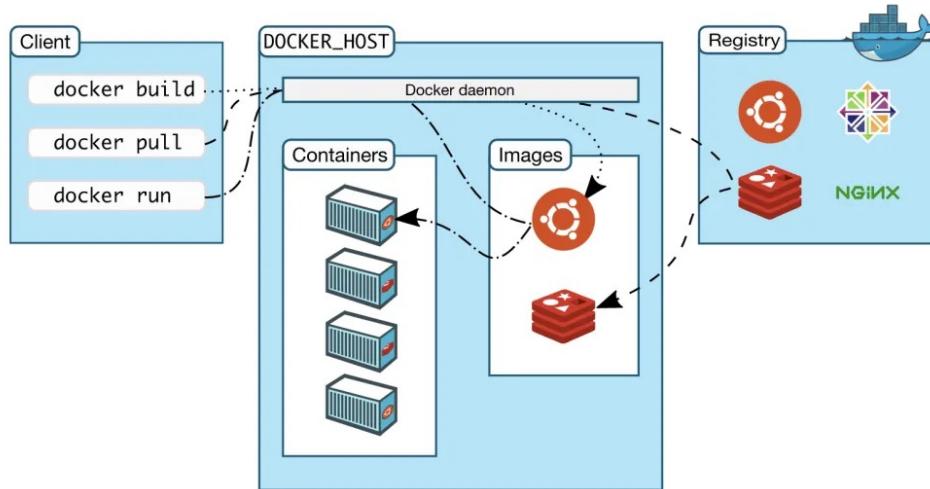


Un approccio diverso si realizza con soluzioni container-based come Docker. Mentre una macchina virtuale astrae l'hardware, i container limitano il loro livello di astrazione al solo sistema operativo con la condivisione dello stesso sistema operativo, con il kernel, la connessione di rete e i file di base.

Le istanze vengono eseguite in uno spazio separato, garantendo una diminuzione di consumo della CPU e dell'overload associato

	Tempo di avvio	Tempo di arresto
Docker Containers	<50ms	<50ms
Virtual Machines	30 - 45 secondi	5 - 10 secondi

::: Esecuzione Isolata (3/4)



Questa forma di virtualizzazione si basa sui daemon per gestire i container e le immagini delle applicazioni. Gli attacchi per compromettere gli ambienti di esecuzione sono tipicamente basati su vulnerabilità di questi demoni che presentano bug software o una cattiva configurazione.

CVE Details
The ultimate security vulnerability datasource

Log In Register

Home [Browse](#) » [Docker](#) : Security Vulnerabilities

CVSS Scores Greater Than: 0 1 2 3 4 5 6 7 8 9
Sort Results By : CVE Number Descending CVE Number Ascending CVSS Score Descending Number Of Exploits Descending

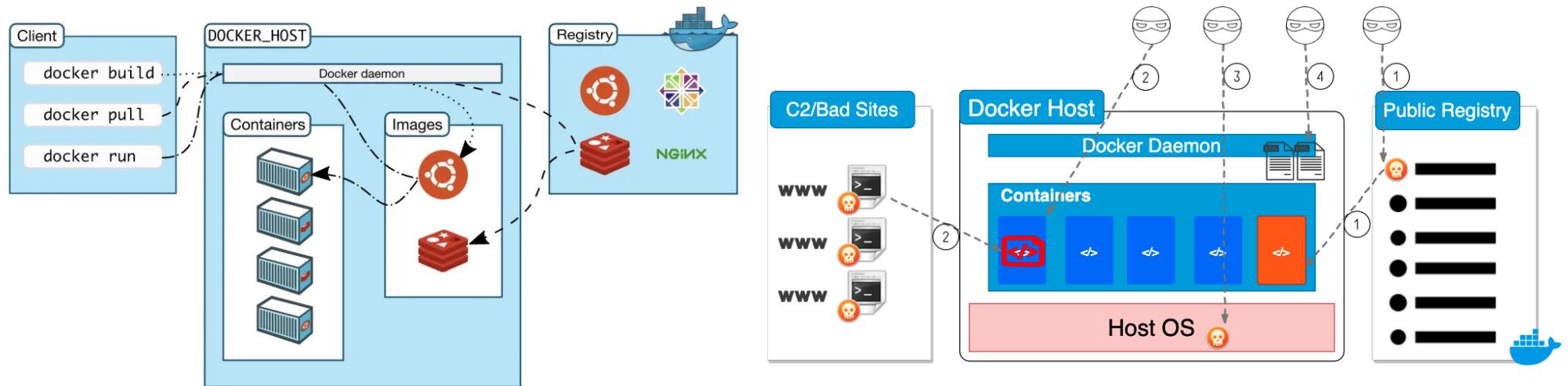
Copy Results Download Results

#	CVE ID	CWE ID	# of Exploits	Vulnerability Type(s)	Publish Date	Update Date	Score	Gained Access Level	Access	Complexity	Authentication	Conf.	Integ.	Avail.
1	CVE-2019-16884	863	Bypass		2019-09-25	2019-10-07	5.0	None	Remote	Low	Not required	None	Partial	None
2	CVE-2019-15752	264	+Priv		2019-08-28	2019-09-04	9.3	None	Remote	Medium	Not required	Complete	Complete	Complete

(e.g.: CVE-2009-1234 or 2010-1234 or 20101234) Search [View CVE](#)

Vulnerability Feeds & Widgets [New](#) [www.itsecdb.com](#)

::: Esecuzione Isolata (3/4)

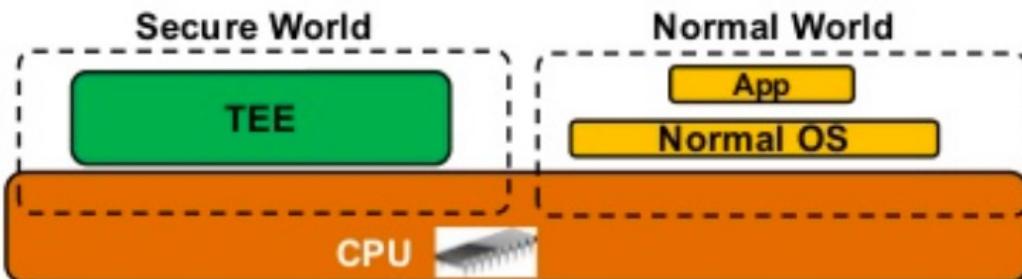


- Deployment di immagini in container con codice dannoso:** le immagini dannose vengono prima inviate a un registro pubblico per essere estratte e distribuite sugli host Docker non protetti.
- Deployment di immagini benigne che scaricano payload dannosi in fase di esecuzione:** le immagini benigne vengono distribuite sugli host Docker, dove i payload dannosi vengono quindi scaricati ed eseguiti.
- Deployment di file dannosi sulla macchina host:** gli avversari montano l'intero file system dell'host su un container e vi accedono dal container.
- Ottenerne informazioni sensibili dal registro Docker:** gli avversari analizzano i registri Docker per trovare informazioni sensibili.

::: Esecuzione Isolata (4/4)

Gli ambienti di esecuzione isolati assistiti da hardware sono stati realizzati per la protezione dei sistemi, e combinano il concetto di esecuzione isolata con tecnologie assistite da hardware. Entrambi sono fondamentali per proteggere i sistemi informatici:

- Il concetto di esecuzione isolata fornisce un TEE (Trusted Execution Environment) per l'esecuzione di strumenti difensivi su un sistema compromesso.
- L'utilizzo di tecnologie assistite da hardware esclude gli hypervisor dal TCB, raggiunge un alto livello di privilegio (ovvero privilegio a livello hardware) e riduce il sovraccarico delle prestazioni, consentendo ai cambi di contesto di essere eseguiti più velocemente nell'hardware.



Separa le applicazioni in contesti normali e sicuri, dove nel secondo è eseguito il software critico.

::: Ambiente di Esecuzione Sicura

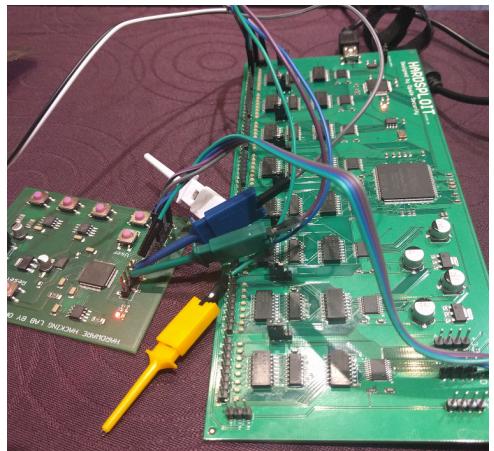
Un ambiente sicuro consente l'archiviazione e l'esecuzione sicure delle applicazioni.

- Esecuzione isolata: ogni applicazione dovrebbe essere eseguita indipendentemente dalle altre applicazioni.
 - un'applicazione dannosa non può accedere ai dati sensibili conservati da altre applicazioni protette nella memoria.
 - l'applicazione dannosa non può accedere al codice o ai dati di un'applicazione mentre è in esecuzione e inoltre non può alterarne l'esecuzione.
- Archiviazione sicura: sono garantite l'integrità e la segretezza di tutti i dati, inclusi i file binari che rappresentano le applicazioni da eseguire. i dati più sensibili da proteggere risultano essere le password, le chiavi di crittografia e i certificati.
- Provisioning sicuro: questa proprietà garantisce sia la capacità di inviare in modo sicuro i dati a un software specifico nell'ambiente protetto sia la capacità di installare in remoto applicazioni sensibili e trasferire chiavi di crittografia o certificati.

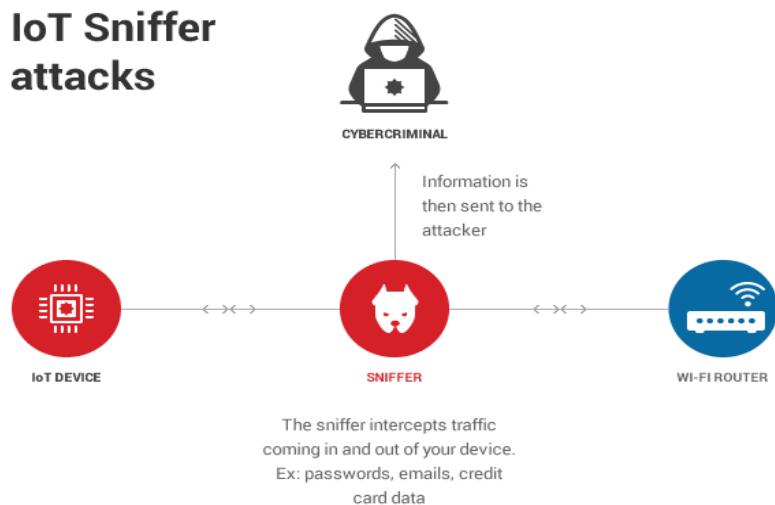
Un'applicazione eseguita in un ambiente che garantisce queste tre caratteristiche è chiamata applicazione sicura.

::: Soluzioni Hardware

Le soluzioni di sicurezza hardware hanno il vantaggio di ridurre notevolmente le intrusioni e gli attacchi.



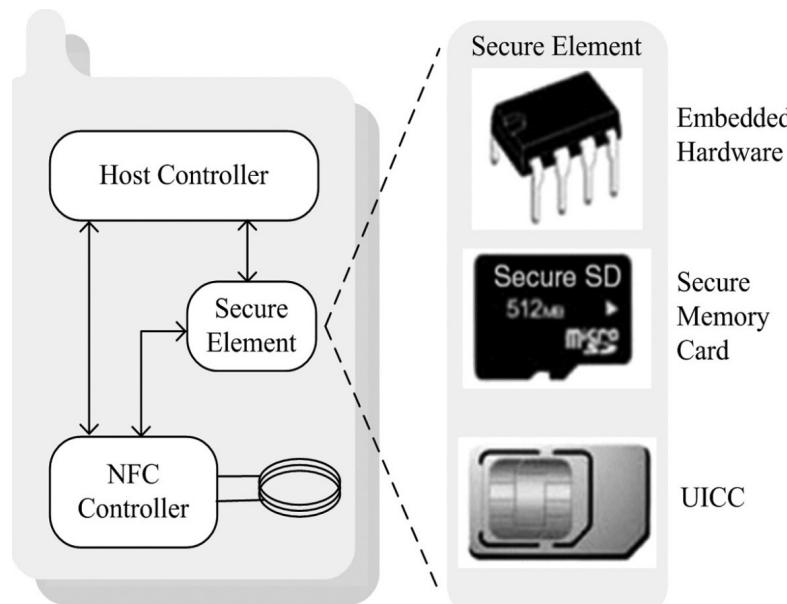
Gli avversari dannosi possono eseguire attacchi in grado di ispezionare la memoria di archiviazione o intercettare la chiave durante il processo di esecuzione, rendendo la crittografia non completamente in grado di affrontare i problemi di sicurezza nel mondo IoT.



La massiccia implementazione della scheda SIM che incorpora un Secure Element è l'origine delle soluzioni basate su hardware.

::: Elementi Sicuri

Un Secure Element (SE) è un chip a microprocessore in grado di memorizzare dati sensibili ed eseguire app sicure come i pagamenti. Agisce come un vault, proteggendo ciò che è all'interno dell'SE (applicazioni e dati) dagli attacchi malware tipici dell'host (ovvero il sistema operativo del dispositivo).

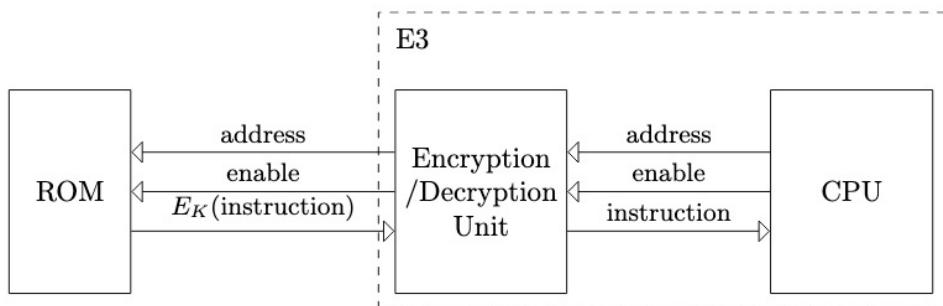


Solo le applicazioni firmate dal produttore possono essere eseguite in un SE, il che limita le possibilità di un utente malintenzionato di app e di un root attacker. Questa è una grande limitazione per gli sviluppatori e il motivo principale che ha ostacolato lo sviluppo di questa tecnologia.

::: Ambiente di Esecuzione Criptato

Un Encrypted Execution Environment (E3) è un ambiente di esecuzione in cui il software è crittografato e consente l'esecuzione senza rivelare le istruzioni che compongono l'applicazione.

È probabile che venga impiegata una società che assegna una grande quantità di valore monetario e tempo per proteggere i propri investimenti dalla contraffazione e da altre duplicazioni non autorizzate.



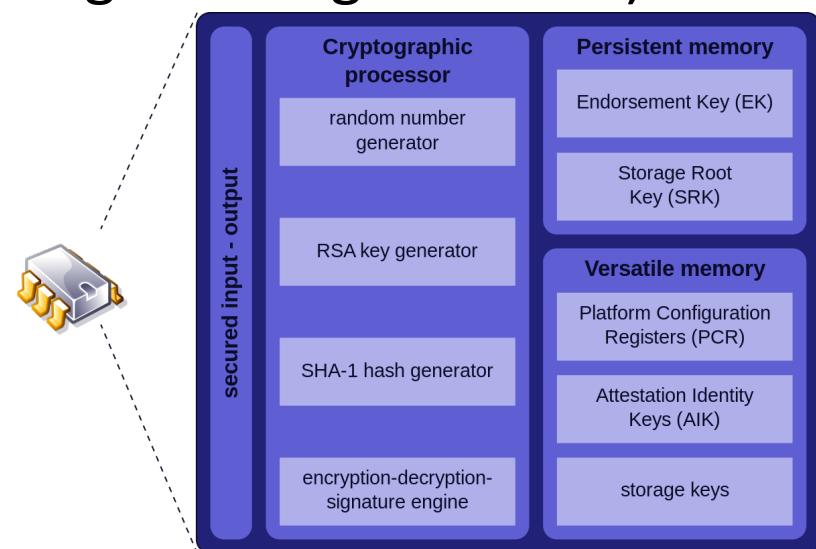
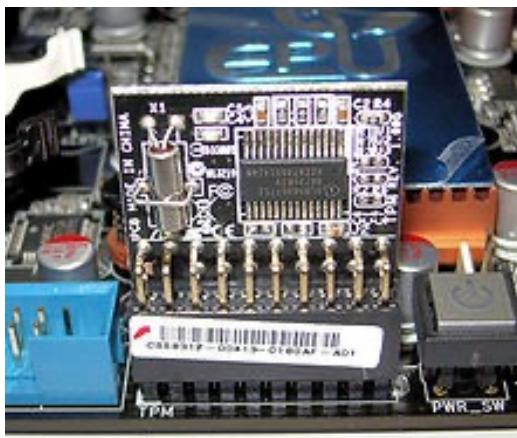
Se ogni dispositivo fornito con il software ha la propria chiave E3, il software E3 per un dispositivo non verrà eseguito su un altro.

Ciò non implica necessariamente che le istruzioni possano essere eseguite direttamente dal loro stato crittografato. In effetti, è accettabile che l'E3 decifri ed esegua le istruzioni fintanto che le istruzioni in testo normale non vengono rivelate esternamente.

::: Trusted Platform Module (1/4)

Il TPM è costituito da un microcontrollore con funzionalità crittografiche aggiuntive. Il TPM può essere associato al dispositivo utilizzando un bus LPC (Low Pin Count) e può eseguire operazioni crittografiche molto complesse dalla crittografia simmetrica alla crittografia asimmetrica RSA.

Il vantaggio dell'utilizzo di un TPM è che lo sviluppatore non deve sapere nulla sull'implementazione di questi algoritmi, poiché il TPM fornisce un'API (Application Programming Interface).



::: Trusted Platform Module (2/4)

Le funzionalità crittografiche di un TPM sono le seguenti:

- Acceleratore RSA: un modulo motore esegue le operazioni crittografiche RSA con una lunghezza massima della chiave di 2048 bit, ed è utilizzato durante le operazioni di firma digitale e wrapping delle chiavi.
- Il TPM è in grado di calcolare valori hash di piccole porzioni di dati, come chiavi crittografiche e certificati, ma non è sufficiente per grandi quantità di dati.
- Generazione di numeri pseudo casuali: questa funzione è molto importante e utile per generare chiavi di crittografia, ad esempio per RSA.

Il TPM è caratterizzato dalla chiave di endorsement, creata casualmente sul chip al momento della produzione, non può essere modificata e non lascia mai il chip, mentre la chiave pubblica viene utilizzata per l'attestazione e per la crittografia dei dati sensibili inviati al chip.

::: Trusted Platform Module (3/4)

Le chiavi di identità dell'attestazione vengono conservate dal TPM per eseguire un'autenticazione con un provider di servizi.

Il TPM archivia tre tipi di certificati:

- Il certificato di verifica dell'autenticità garantisce l'integrità della chiave di verifica dell'autenticità. Questo certificato può essere fornito dallo stesso emittente dell'EK ma non è obbligatorio.
- Il certificato della piattaforma viene fornito dal fornitore della piattaforma e garantisce che tutti i componenti di sicurezza forniti con la piattaforma siano autentici. Questo certificato abilita l'attendibilità della piattaforma.
- Il certificato di conformità viene fornito da un laboratorio di valutazione di terze parti o dallo stesso fornitore della piattaforma. Attesta che le proprietà di sicurezza dichiarate dal produttore sono autentiche.

::: Trusted Platform Module (4/4)

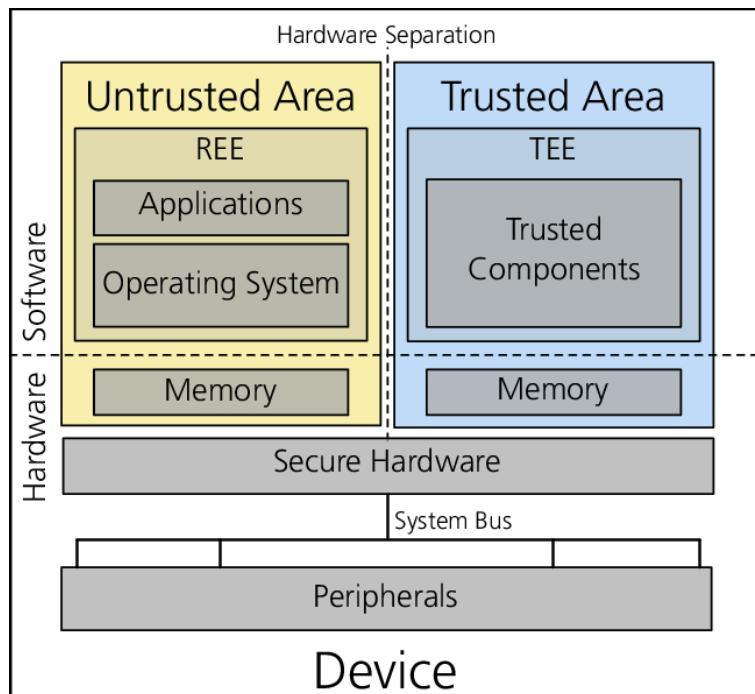
Il TPM protegge i dati mediante due possibili soluzioni:

- **Memory curtaining** estende le comuni tecniche di protezione della memoria per fornire un isolamento completo delle aree sensibili della memoria, anche al sistema operativo, ad esempio le posizioni contenenti chiavi crittografiche.
- **Sealed storage** protegge le informazioni private legandole alle informazioni di configurazione della piattaforma. I dati possono essere rilasciati utilizzando una chiave che deriva dallo stato del sistema, ovvero il software utilizzato e l'hardware su cui è in esecuzione. Le informazioni possono essere utilizzate solo con la stessa combinazione di software e hardware.

Come caso d'uso, un TPM può essere utilizzato come PKI che fornisce le chiavi e i certificati necessari per stabilire comunicazioni protette e firmare documenti.

::: Trusted Execution Environment (1/5)

TEE combina parti hardware e software e consentono di dividere il sistema in due ambienti di esecuzione.



- Il Rich Execution Environment (REE), è un sistema operativo tradizionale con notevolmente la superficie di attacco.
- Il TEE rappresenta il sistema operativo sicuro responsabile dell'esecuzione di operazioni sensibili. Ha anche la capacità di proteggere il display e l'ingresso utilizzando una modalità sicura dei bus che collegano il processore alle periferiche I/O.

Il TEE divide il processore in due zone con un sistema operativo sicuro e altri meccanismi per migliorare la sicurezza dell'elaborazione dei dati sensibili.

::: Trusted Execution Environment (2/5)

L'altra caratteristica essenziale è l'archiviazione sicura, in cui i dati critici e sensibili sono isolati dai dati dell'utente per migliorarne la sicurezza. In opposizione all'SE, il TEE fornisce un canale di comunicazione sicuro tra il processore e la periferica esterna, in particolare l'ingresso e il display.

- Se un'applicazione dannosa può intercettare l'input, può avere accesso a dati sensibili. Anche la visualizzazione sicura è primordiale perché l'utente deve essere sicuro che ciò che vede sullo schermo sia realmente inviato dal mondo sicuro.

È inoltre necessario un processo di avvio sicuro:

1. Leggere una ROM affidabile (bloccata in produzione),
2. Controllo della firma e dell'integrità del sistema operativo sicuro,
3. Configurazione del sistema operativo sicuro.

TEE ha anche una terza modalità chiamata Monitor, utilizzata per eseguire il salvataggio del contesto e il passaggio tra Rich e Secure OS.

::: Trusted Execution Environment (3/5)

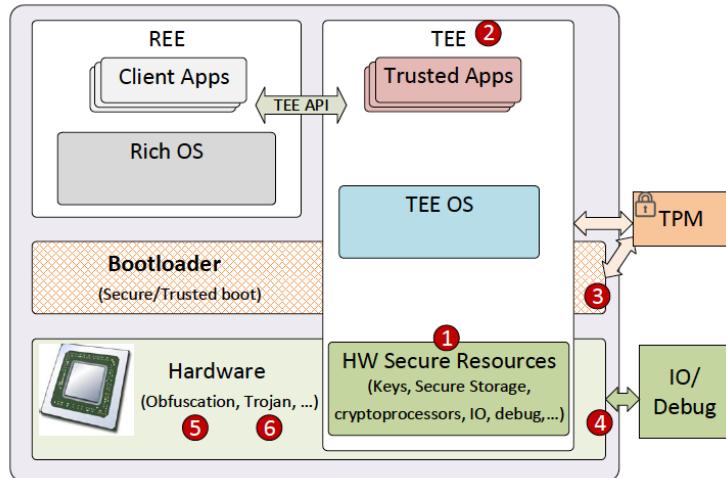
Il sistema operativo sicuro ha un set di istruzioni limitato, necessario per ridurre la superficie di attacco, e pianifica le applicazioni sensibili in esecuzione su di esso per realizzare istruzioni sicure come operazioni crittografiche: generazione di chiavi, criptatura/decrittatura di dati, generazione di firme.

Comparison between the hardware solutions.

Criteria	SE	TEE	TPM
Tamper resistance	✓		✓
Secure input and display		✓	
High computation power		✓	✓
High storage capacity		✓	
Dependency to manufacturer	✓	✓	✓
Proven security level	✓	✓	✓

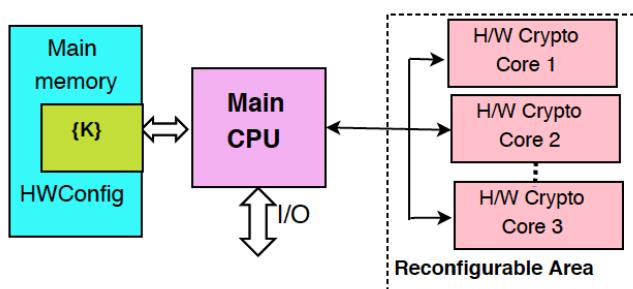
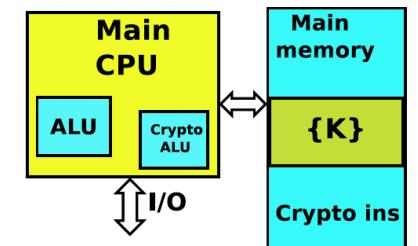
SE e il TPM hanno più funzionalità di sicurezza fisica rispetto a TEE. Tuttavia, TEE fornisce una maggiore potenza di calcolo che consente modelli di sicurezza più complessi.

::: Trusted Execution Environment (4/5)



La base per la realizzazione a livello hardware di TEE è costituita da coprocessore, che sono un processore specializzato che esegue algoritmi crittografici a livello hardware e accelera il processo di cifrature/decifratura per una migliore sicurezza dei dati e protezione delle chiavi segrete.

Un processore generico (GPP) viene personalizzato e utilizzato per implementare alcuni algoritmi crittografici. La personalizzazione principale sono le estensioni del set di istruzioni (possono essere chiamate istruzioni crittografiche) per applicazioni crittografiche. In questo caso, le chiavi segrete vengono salvate nella memoria principale e utilizzate come altri normali dati



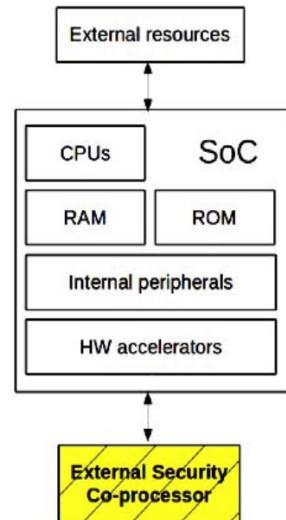
Un coprocessore crittografico è un modulo esterno al GPP che accelera i calcoli crittografici. Le chiavi segrete normalmente non vengono memorizzate nella memoria del coprocessore, ma nei registri dati del processore o nella memoria principale. Il coprocessore crittografico può essere controllato utilizzando il GPP host.

::: Trusted Execution Environment (5/5)

Ci sono molti modi in cui questi TEE possono essere realizzati.

::: Trusted Execution Environment (5/5)

Ci sono molti modi in cui questi TEE possono essere realizzati.



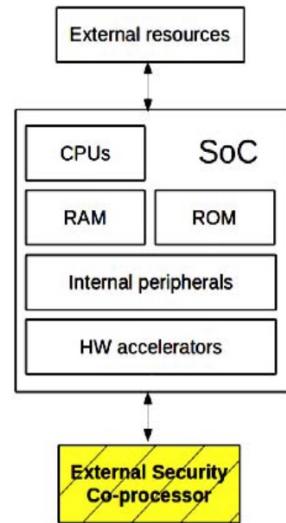
La prima realizzazione utilizza un coprocessore di sicurezza per scaricare le attività critiche per la sicurezza dall'ambiente operativo principale.

- I vantaggi sono che l'operazione può essere generalmente completamente isolata e può essere eseguita simultaneamente con il nucleo principale.
- Lo svantaggio è che c'è un sovraccarico associato al trasferimento dei dati da e verso il core.

Può avere un coprocessore di sicurezza esterno (con HSM) o integrato (TPM).

::: Trusted Execution Environment (5/5)

Ci sono molti modi in cui questi TEE possono essere realizzati.



La prima realizzazione utilizza attività critiche per la sicurezza

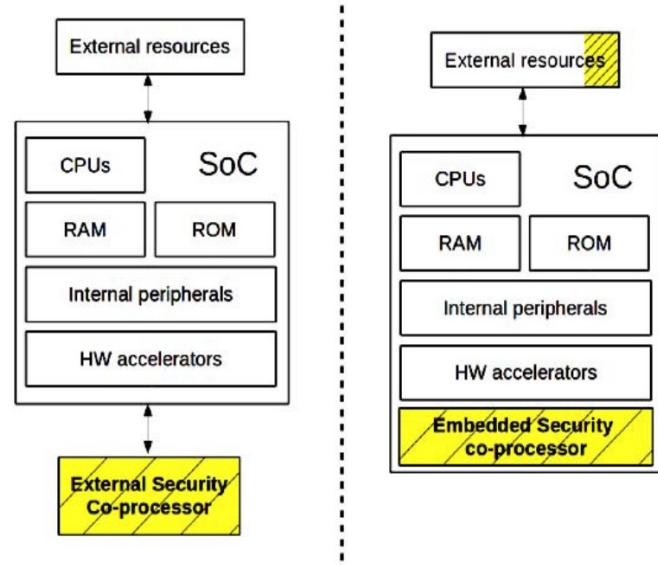
- I vantaggi sono che è completamente isolata dal nucleo principale.
- Lo svantaggio è che c'è una connessione da e verso il core.

Un TPM è un chip hardware sulla scheda madre ed è in grado di fornire la crittografia completa dei dischi. TPM mantiene i dischi rigidi bloccati fino a quando il sistema non completa una verifica del sistema o un processo di autenticazione. D'altra parte, HSM è un dispositivo rimovibile o esterno che genera, archivia e gestisce chiavi crittografiche.

Può avere un **coprocessore di sicurezza esterno** (con HSM) o integrato (TPM).

::: Trusted Execution Environment (5/5)

Ci sono molti modi in cui questi TEE possono essere realizzati.

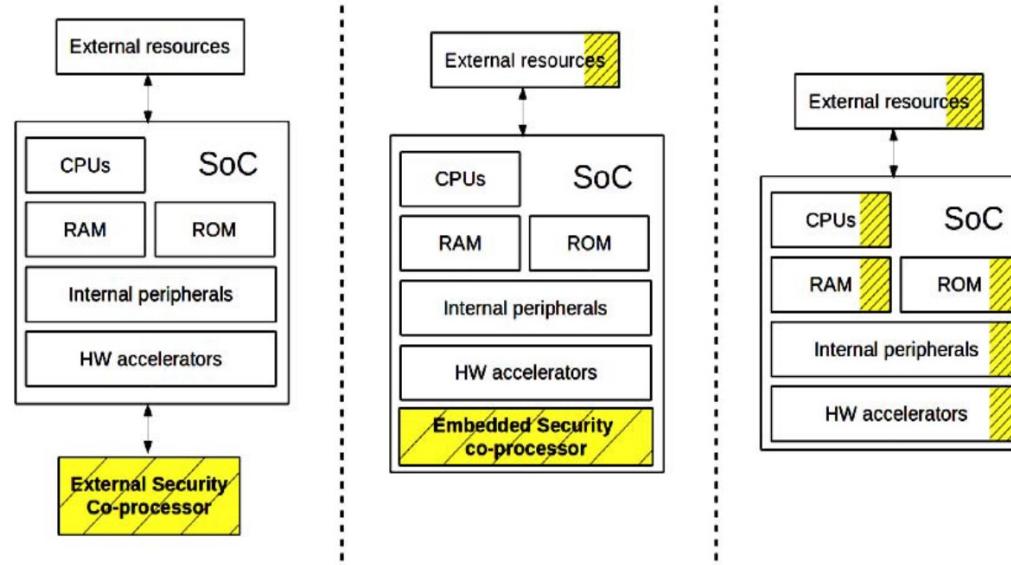


Molte famose architetture TEE supportano un nuovo tipo di configurazione in cui un singolo core supporta più core virtuali che si escludono a vicenda, ovvero quando uno è in esecuzione l'altro è sospeso. Normalmente questa transizione da uno stato all'altro viene eseguita da una sorta di monitor / trigger. Questa configurazione viene talvolta definita "ambiente protetto del processore".

Esempi sono ARM TrustZone e Intel Trusted Execution Technology.

::: Trusted Execution Environment (5/5)

Ci sono molti modi in cui questi TEE possono essere realizzati.



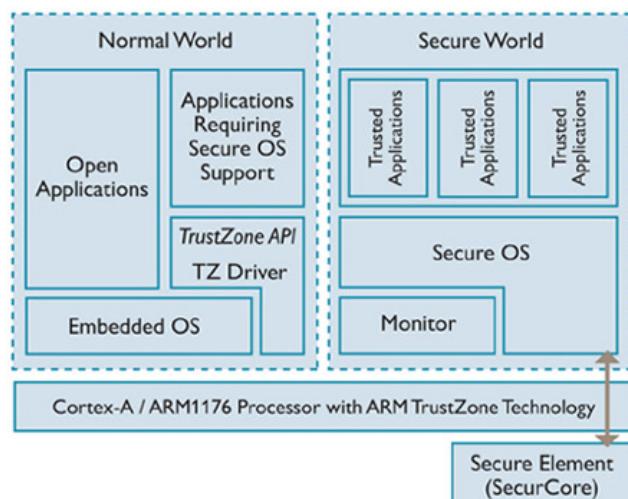
Oltre alle tecniche basate sull'hardware, sono stati presentati altri approcci, come XOM (eXecute-Only-Memory) a livello di architettura. Tali piattaforme realizzano il TEE per separazione architettonica, previene la fuoriuscita di informazioni dalle applicazioni XOM utilizzando il compartmento, dove un compartmento è un contenitore logico che impedisce alle informazioni di entrare o uscire da esso.

Ulteriori esempi di software TEE includono i contenitori Docker.

::: Processori ARM TrustZone (1/2)

Il processore ARM TrustZone ha un'istruzione specifica chiamata Secure Monitor Call (SMC) da invocare durante l'esecuzione nel mondo normale per entrare in "modalità monitor" che esegue la transizione al mondo sicuro.

- Non è necessario scaricare i dati da / verso il mondo sicuro. È previsto un costo aggiuntivo per memorizzare / ripristinare lo stato del dispositivo all'ingresso / uscita da una determinata modalità.
- Lo svantaggio è che quando un mondo è attivo, l'altro deve essere completamente fermato, complicando così la gestione delle interruzioni.



L'innovazione architettonica più significativa è l'aggiunta di un nuovo bit del processore, il 33-esimo, che indica in quale modo il processore è attualmente in esecuzione.

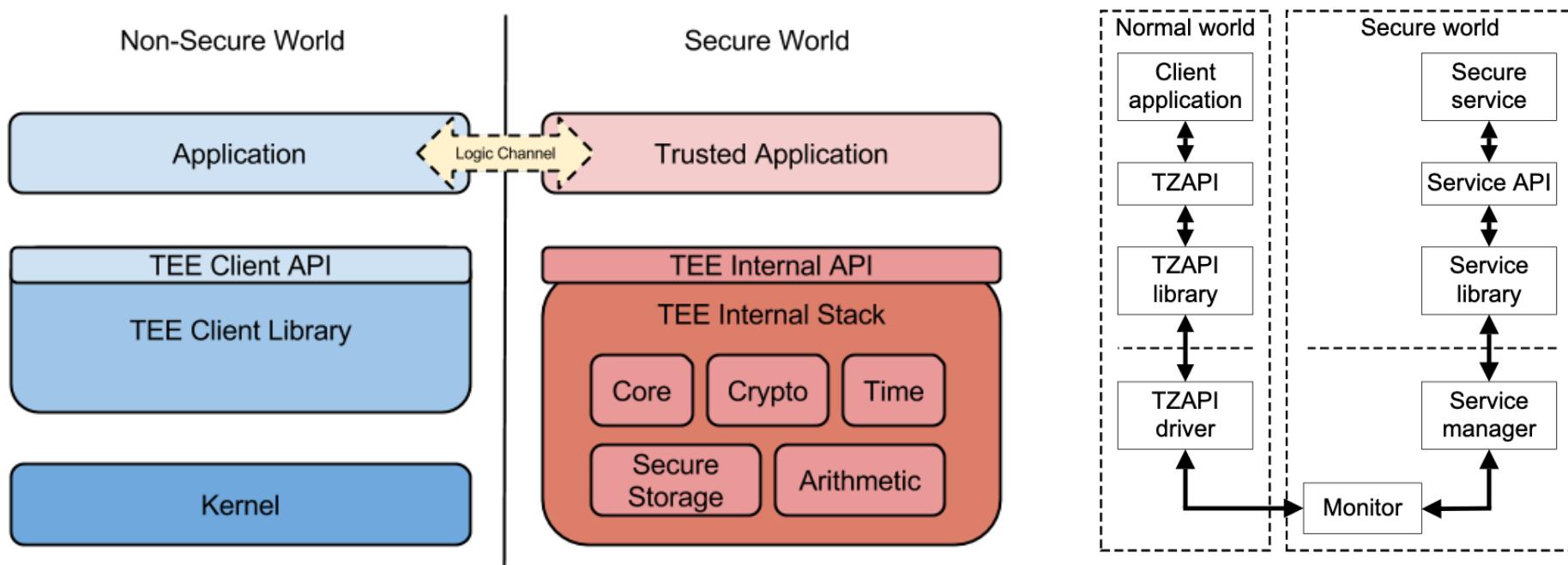
Il TrustZone address space controller (TZASC) estende la sicurezza a livello di memoria, abilitando la partizione in differenti regioni.

L'unità di gestione della memoria (MMU) che riconosce la zona di fiducia fornisce due interfacce MMU distinte.

::: Processori ARM TrustZone (2/2)

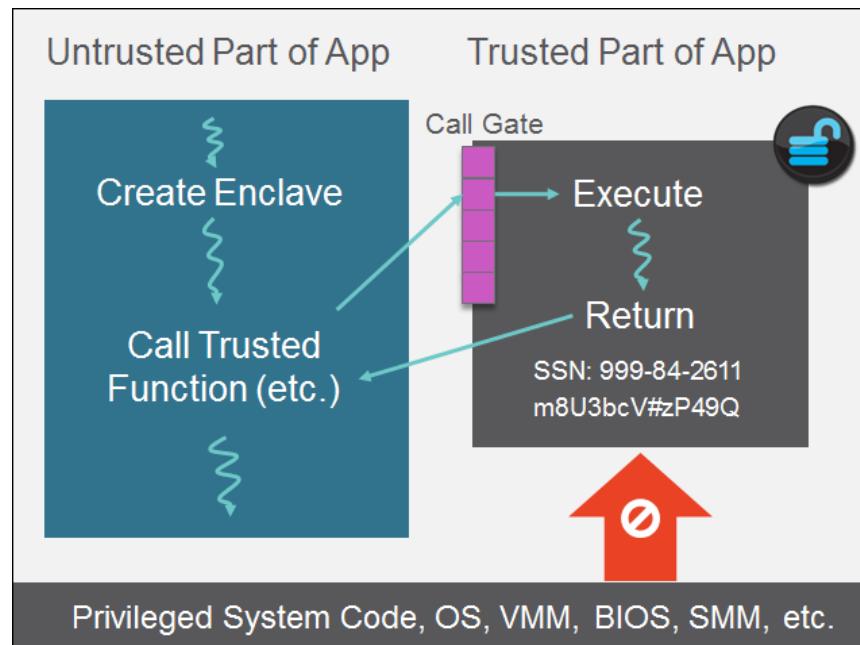
L'API TrustZone (TZAPI) specifica in che modo le applicazioni non protette (NSA), in esecuzione nell'ambiente ricco, interagiscono con l'ambiente di esecuzione isolato.

Seguendo un modello client-server, l'API definisce un insieme di interfacce software astratte mediante le quali un NSA può interagire con un TA. L'API consente ai client di inviare comandi e richieste a un TA e scambiare dati tra i due mondi. L'API TrustZone non include alcuna specifica su come sviluppare applicazioni in esecuzione all'interno dell'ambiente di esecuzione isolato.



::: Intel SGX (1/3)

Tecniche come Intel Software Guard Extensions (Intel SGX) utilizzano sia hardware che software per implementare TEE.



Le applicazioni Intel SGX sono costituite da una parte affidabile e una parte non attendibile. Quando l'applicazione deve lavorare con un segreto, crea un'enclave che viene collocata nella memoria attendibile. Quindi chiama una funzione attendibile per entrare nell'enclave, dove gli viene fornita una visualizzazione dei suoi segreti in testo chiaro.

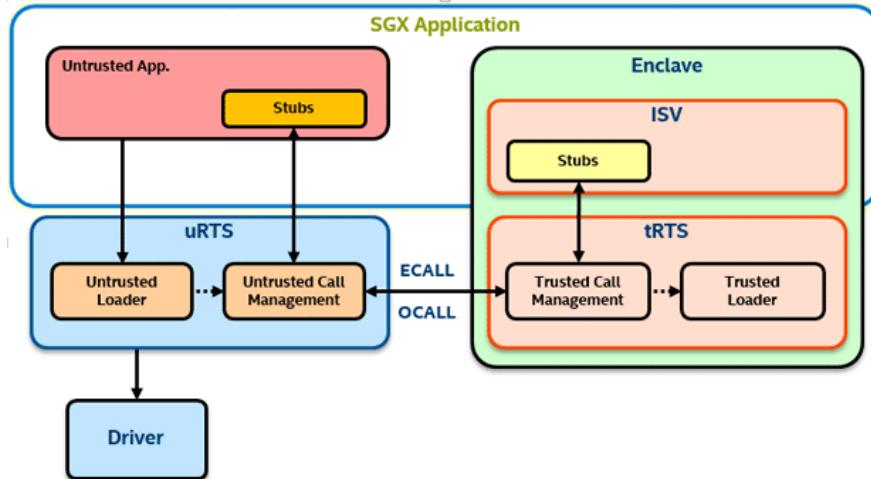
Tutti gli altri tentativi di accesso alla memoria dell'enclave dall'esterno dell'enclave sono negati dal processore, anche quelli effettuati da utenti privilegiati. Ciò impedisce che i segreti nell'enclave vengano svelati.

::: Intel SGX (2/3)

SGX è un insieme di istruzioni della CPU che consentono a un'applicazione di istanziare un contenitore protetto, denominato enclave.

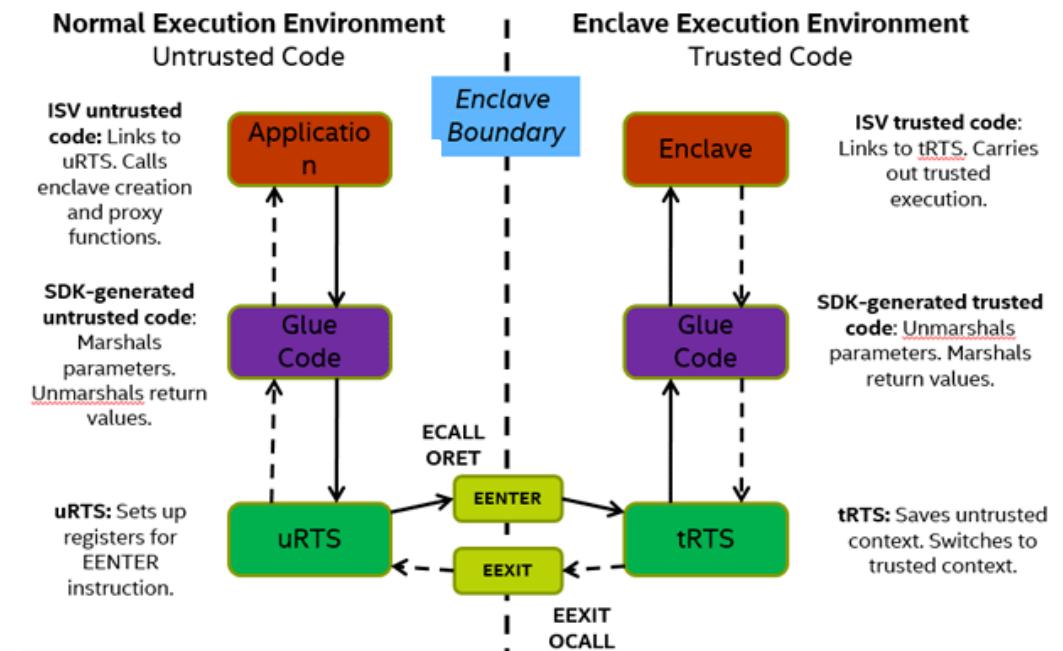
- Un'enclave è definita come un'area protetta nello spazio degli indirizzi dell'applicazione che non può essere alterata da codice esterno all'enclave, nemmeno da codice con privilegi superiori.
- Il core non esegue una transizione completa, ma parti di un'applicazione standard sono protette da meccanismi hardware nel core.
- I vantaggi sono che non è necessario trasferire i dati avanti e indietro tra i core o impostare transizioni complicate da e verso un mondo sicuro e non è necessario un ambiente operativo separato come richiesto in altri stili di configurazione TEE.
- Il componente affidabile dovrebbe essere il più piccolo possibile: una grande enclave con un'interfaccia complessa non consuma solo più memoria protetta: crea anche una superficie di attacco più ampia.
- Le enclave dovrebbero anche avere un'interazione minima tra componenti attendibili e non attendibili: limitare queste dipendenze rafforzerà l'enclave contro gli attacchi.

::: Intel SGX (3/3)



Trusted Run-Time System (tRTS) rappresenta il codice che viene eseguito all'interno dell'ambiente dell'enclave ed esegue la ricezione di chiamate (ECALL) dall'applicazione e l'esecuzione di chiamate all'esterno (OCALL) dell'enclave o la gestione dell'enclave stessa.

Untrusted Run-Time System (uRTS) indica il codice che viene eseguito al di fuori dell'ambiente enclave ed esegue il caricamento e la manipolazione di un'enclave, l'esecuzione di chiamate (ECALL) a un'enclave e la ricezione di chiamate (OCALL) da un'enclave.



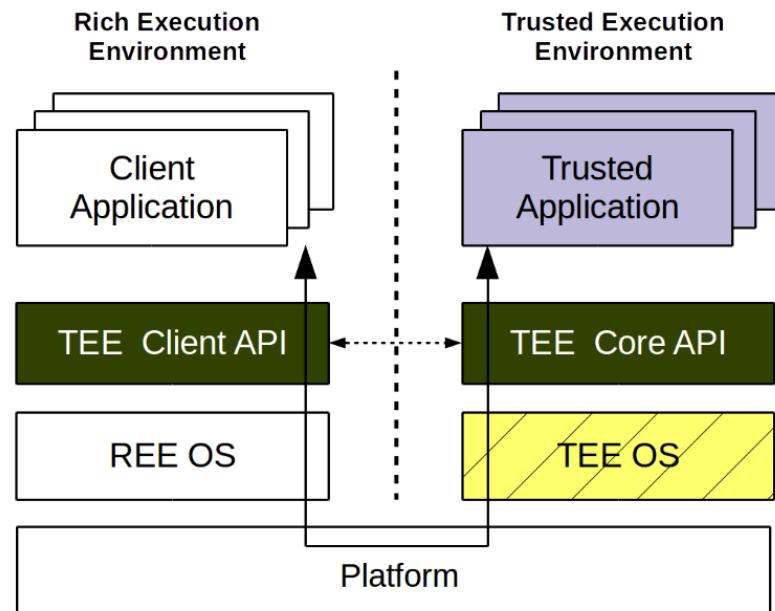
::: Open-TEE (1/3)

Agli sviluppatori di applicazioni mancano le interfacce per utilizzare le funzionalità TEE basate su hardware. In effetti, il loro utilizzo è stato limitato principalmente alle applicazioni sviluppate dai fornitori di dispositivi. La recente standardizzazione delle interfacce TEE da parte di Global Platform (GP) promette di risolvere parzialmente questo problema consentendo alle applicazioni conformi a GP di essere eseguite su TEE di diversi fornitori.

Gli sviluppatori ordinari che desiderano sviluppare applicazioni affidabili devono affrontare sfide significative. È difficile ottenere l'accesso alle interfacce hardware TEE senza alcun supporto da parte dei fornitori. Gli strumenti e il software necessari per sviluppare ed eseguire il debug di applicazioni affidabili possono essere costosi o inesistenti, con solo tecniche di debug primitive come il "tracciamento di stampa".

Un TEE virtuale chiamato Open-TEE, conforme alle specifiche GP, è stato proposto come TEE virtuale conforme agli standard implementato interamente nel software consentirà agli sviluppatori di creare applicazioni TEE utilizzando strumenti e ambienti di sviluppo con cui hanno già familiarità.

::: Open-TEE (2/3)



La specifica GP è composta da

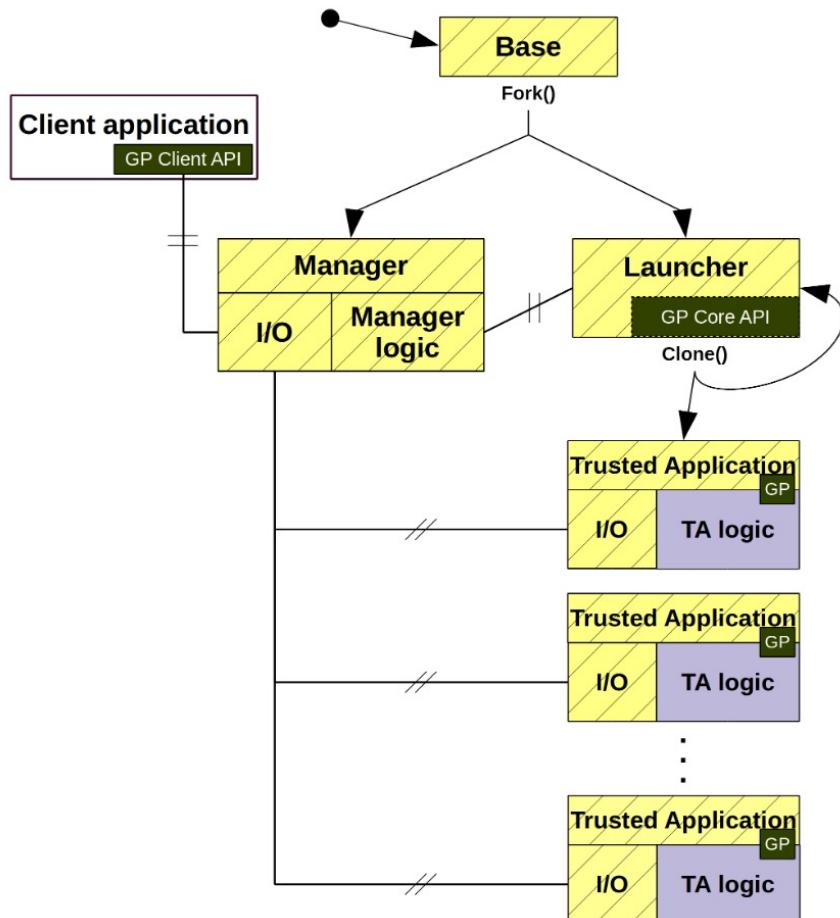
- L'API TEE Core fornisce un ampio set di funzionalità, come un'API crittografica e uno storage sicuro, per implementare un TA.
- L'API client TEE è uno strato molto generico e sottile costituito da un numero limitato di funzioni e definizioni per trasferire i dati avanti e indietro da REE a TA.

Tra l'API client TEE in esecuzione su REE e l'API TEE Core in esecuzione su TEE, abbiamo un'efficace RPC (Remote Procedure Call) in cui un processo in esecuzione in REE può richiamare attività nel TEE.

Questi sforzi di standardizzazione nella piattaforma globale potrebbero risolvere il problema dei TEE interoperabili. Tuttavia, non rimuovono l'ostacolo all'accesso all'hardware richiesto né semplificano il compito di sviluppare e testare le TA.

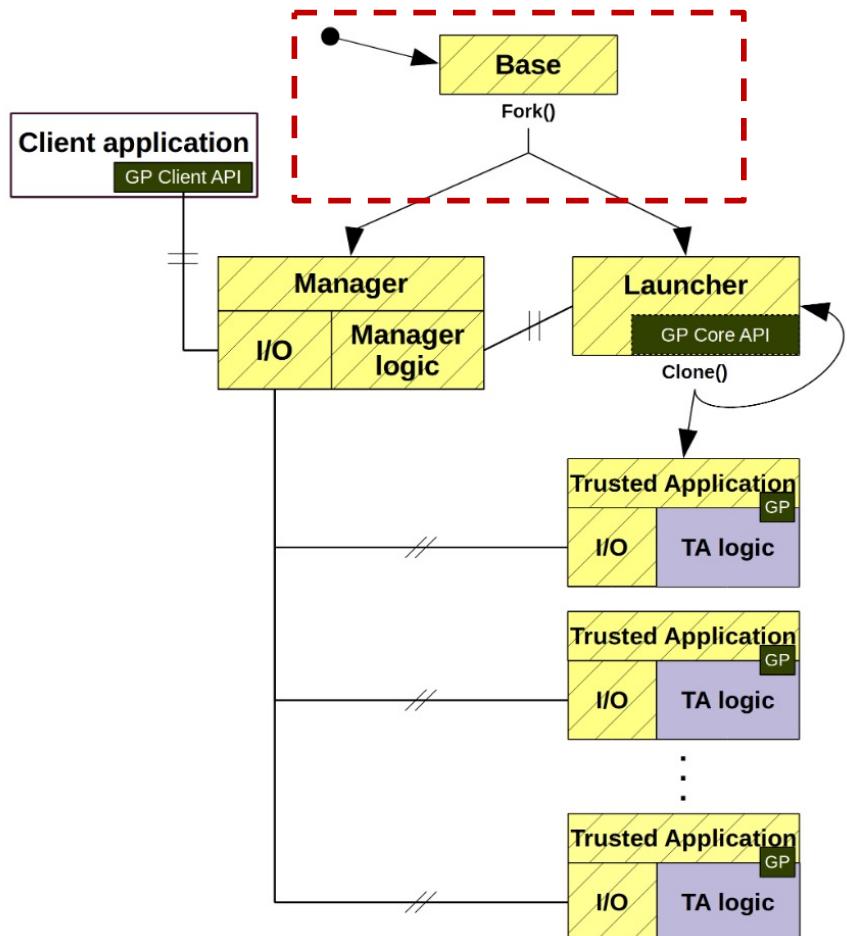
::: Open-TEE (3/3)

Open-TEE fornisce un'architettura e un kit di sviluppo software (SDK) che implementa le specifiche GP come framework e un insieme di strumenti familiari allo sviluppatore, eliminando così la necessità di hardware specializzato e le spese generali che comporta.



::: Open-TEE (3/3)

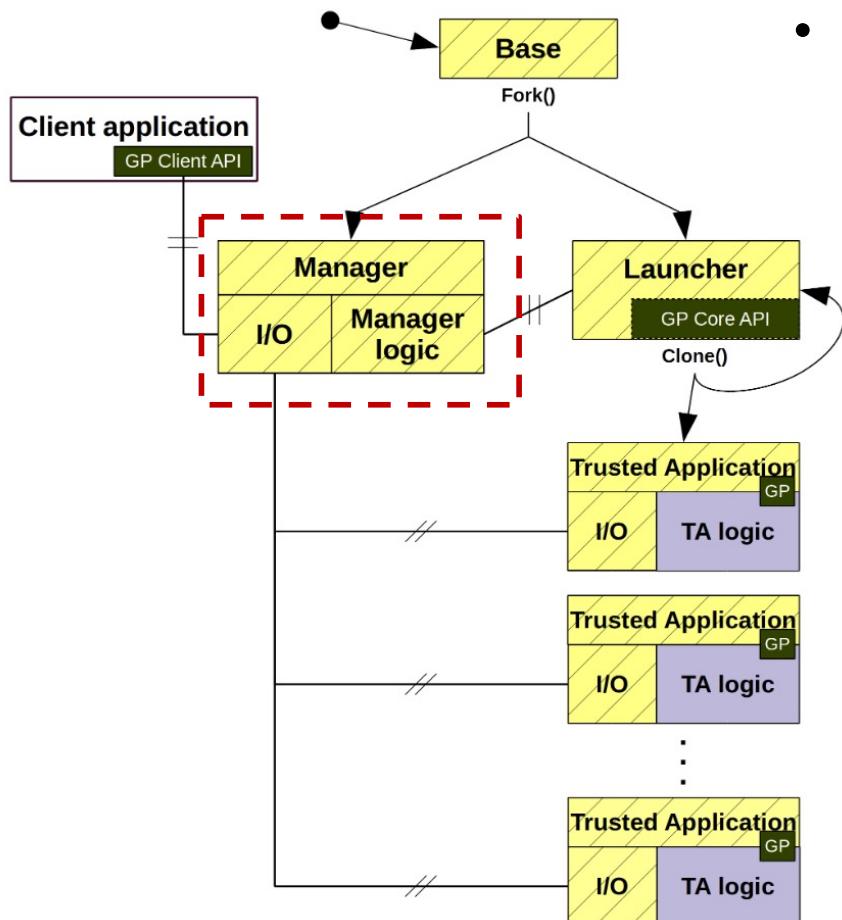
Open-TEE fornisce un'architettura e un kit di sviluppo software (SDK) che implementa le specifiche GP come framework e un insieme di strumenti familiari allo sviluppatore, eliminando così la necessità di hardware specializzato e le spese generali che comporta.



- Open-TEE è progettato per funzionare come un processo demone nello spazio utente. Inizia l'esecuzione di Base, un processo che incapsula la funzionalità TEE nel suo complesso. Una volta inizializzato, Base eseguirà il fork per creare due processi indipendenti ma correlati.

::: Open-TEE (3/3)

Open-TEE fornisce un'architettura e un kit di sviluppo software (SDK) che implementa le specifiche GP come framework e un insieme di strumenti familiari allo sviluppatore, eliminando così la necessità di hardware specializzato e le spese generali che comporta.

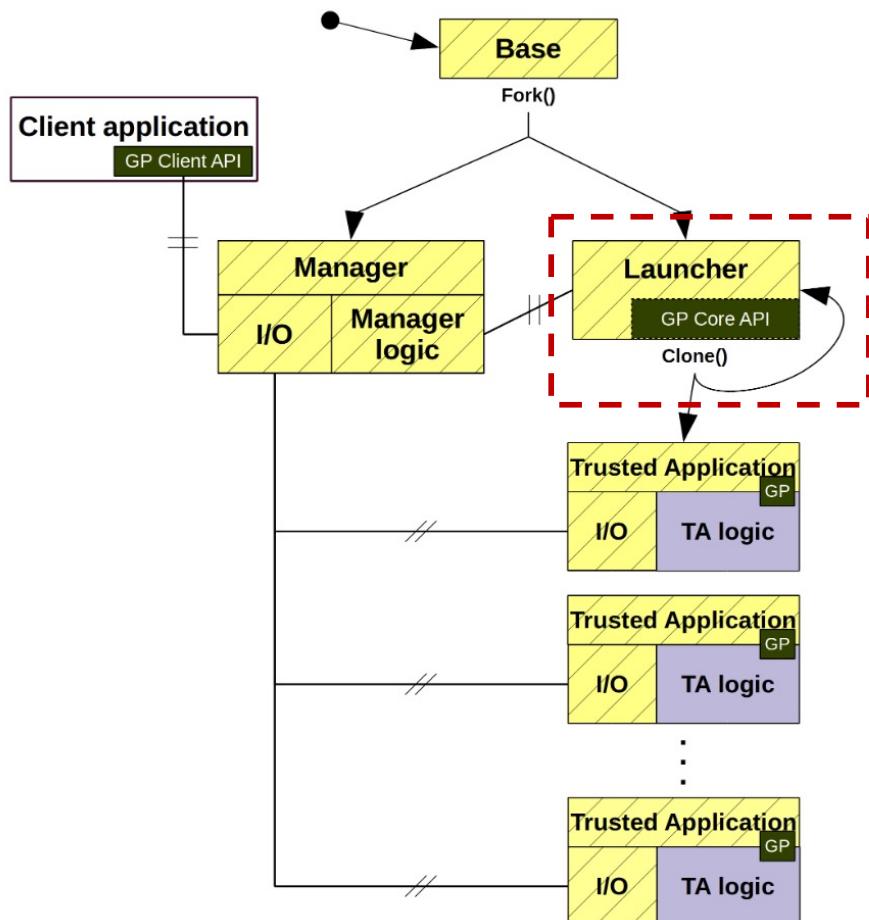


- Manager può essere visualizzato come il "sistema operativo" di Open-TEE:
 1. Gestione delle connessioni tra le applicazioni,
 2. Monitoraggio dello stato del TA,
 3. Fornire archiviazione sicura per TA,
 4. Controllo delle regioni di memoria condivisa per le applicazioni.

La centralizzazione in un processo di controllo può anche essere visto come un wrapper che astrae l'ambiente in esecuzione e lo riconcilia con i requisiti imposti dagli standard GP TEE.

::: Open-TEE (3/3)

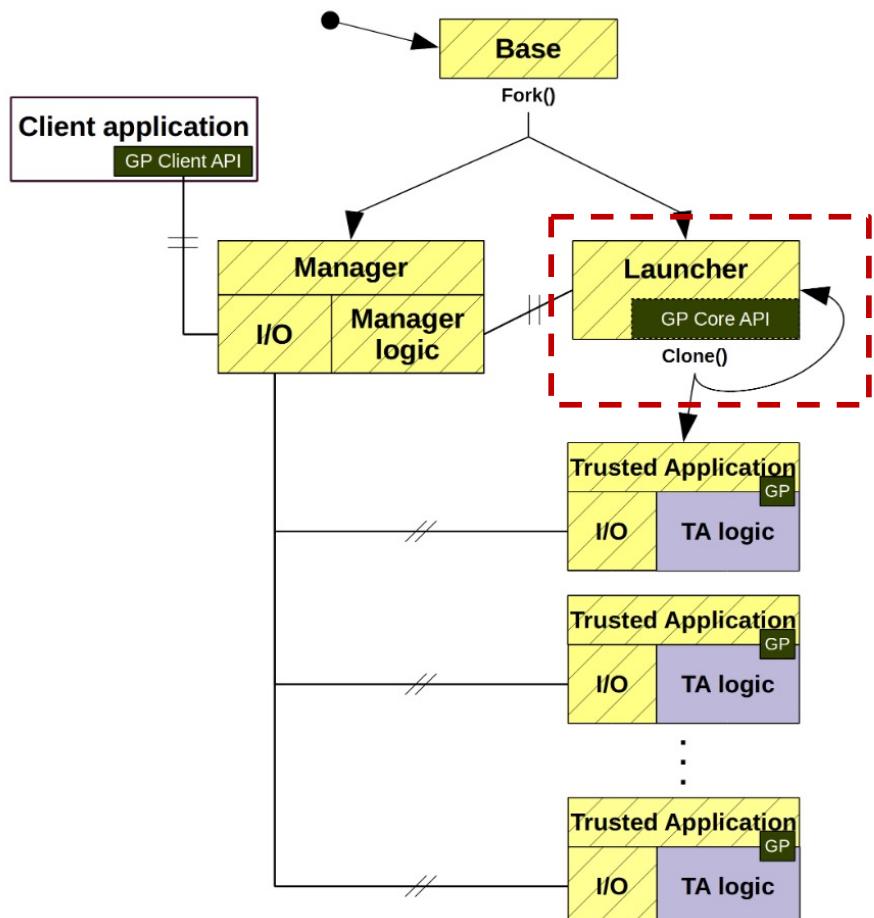
Open-TEE fornisce un'architettura e un kit di sviluppo software (SDK) che implementa le specifiche GP come framework e un insieme di strumenti familiari allo sviluppatore, eliminando così la necessità di hardware specializzato e le spese generali che comporta.



- L'unico scopo di Launcher è creare nuovi processi TA in modo efficiente. Quando viene creato per la prima volta, Launcher caricherà una libreria condivisa che implementa l'API TEE Core e attenderà ulteriori comandi da Manager.
- Manager segnalerà Launcher quando è necessario avviare un nuovo TA.
- Dopo aver ricevuto il segnale, Launcher si clonerà da solo. Il clone caricherà quindi la libreria condivisa corrispondente al TA richiesto.

::: Open-TEE (3/3)

Open-TEE fornisce un'architettura e un kit di sviluppo software (SDK) che implementa le specifiche GP come framework e un insieme di strumenti familiari allo sviluppatore, eliminando così la necessità di hardware specializzato e le spese generali che comporta.



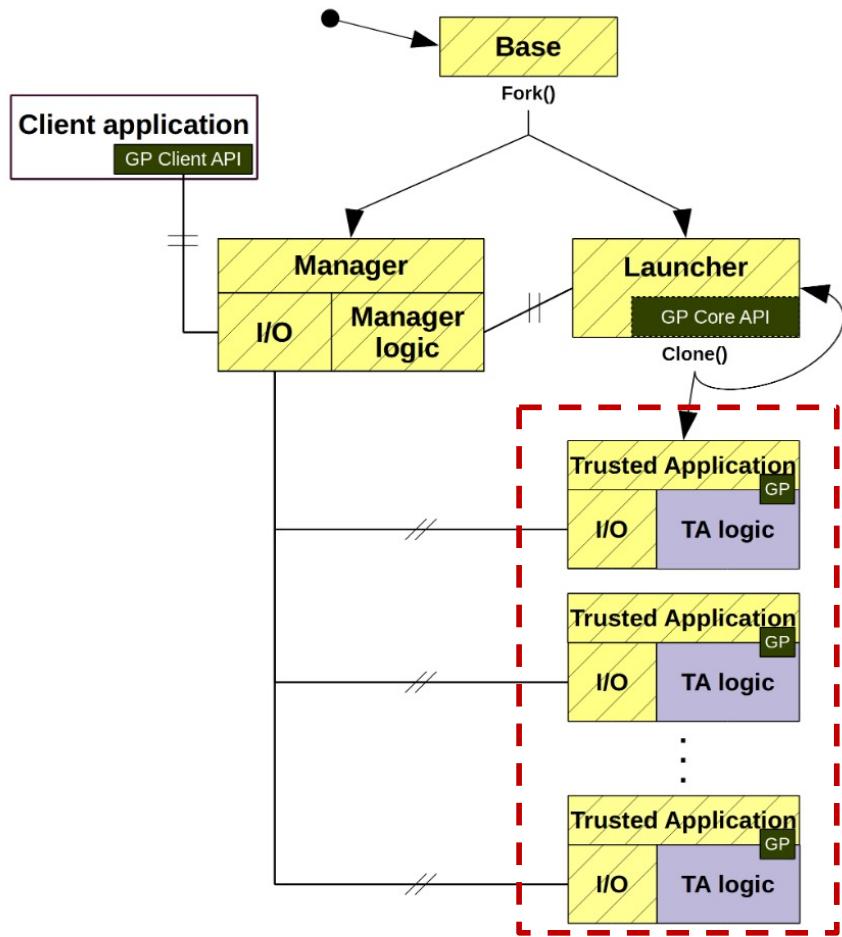
- Il design di Launcher segue il modello di progettazione "zigote".

Zygote è un processo speciale del sistema operativo Android che abilita il codice condiviso su Dalvik / Art VM in contrasto con Java VM in cui ogni istanza ha la propria copia dei file di classe della libreria principale e degli oggetti heap.

- Un processo TA appena creato viene quindi reimpostato su Manager in modo che sia possibile controllarlo.

::: Open-TEE (3/3)

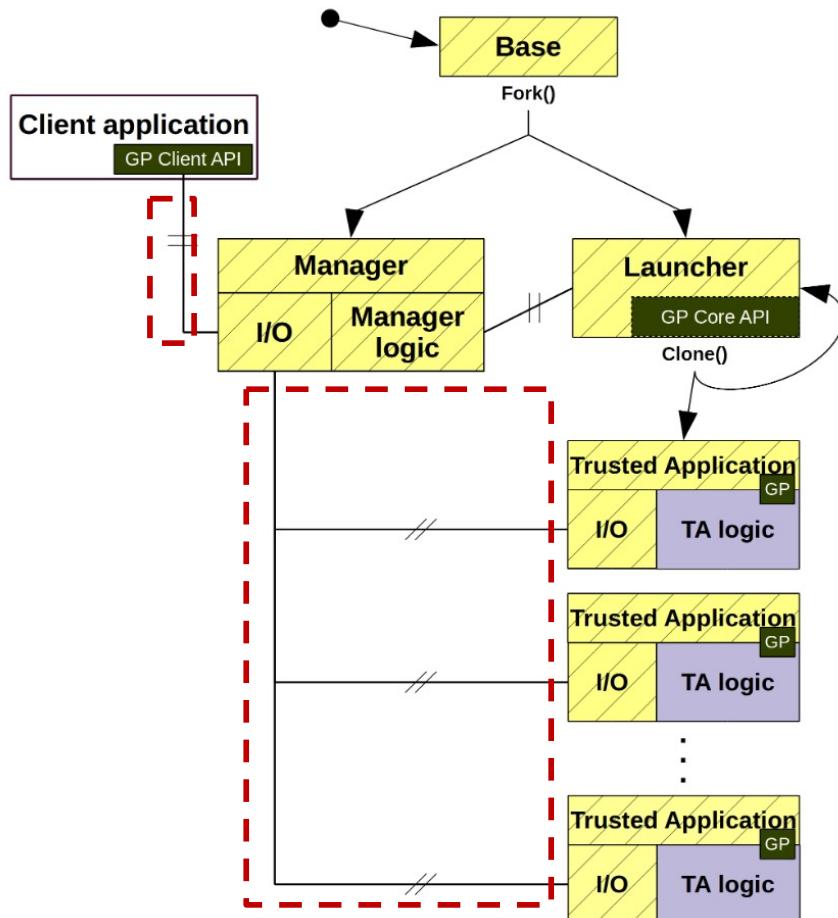
Open-TEE fornisce un'architettura e un kit di sviluppo software (SDK) che implementa le specifiche GP come framework e un insieme di strumenti familiari allo sviluppatore, eliminando così la necessità di hardware specializzato e le spese generali che comporta.



- I processi TA sono stati divisi in due thread. Il primo gestisce Inter-Process Communication (IPC) e il secondo è il thread di lavoro, indicati rispettivamente come thread IO e TA Logic.
- Questo modello architettonico consente di interrompere il processo senza arrestarlo e consente una maggiore separazione e astrazione della funzionalità TA dal framework Open-TEE.

::: Open-TEE (3/3)

Open-TEE fornisce un'architettura e un kit di sviluppo software (SDK) che implementa le specifiche GP come framework e un insieme di strumenti familiari allo sviluppatore, eliminando così la necessità di hardware specializzato e le spese generali che comporta.



- L'API client TEE e l'API TEE Core sono implementate come librerie condivise per ridurre il consumo di codice e memoria.
- Open-TEE implementa un protocollo di comunicazione oltre ai socket del dominio Unix e ai segnali di interelaborazione come mezzo per controllare il sistema e trasferire i messaggi tra CA e TA.

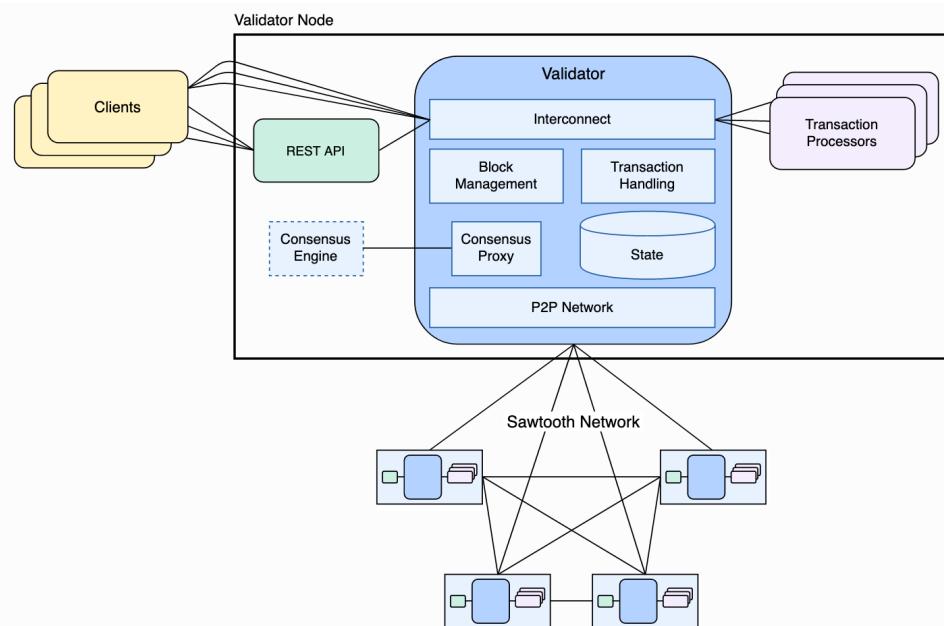


Applicazione di TEE nelle Blockchain

::: Hyperledger Sawtooth (1/4)

Hyperledger Sawtooth è una piattaforma blockchain aziendale altamente modulare che consente alle applicazioni di scegliere le regole di transazione, i permessi e gli algoritmi di consenso che supportano le loro esigenze aziendali uniche.

Sawtooth semplifica lo sviluppo e la distribuzione di un'applicazione fornendo una chiara separazione tra il livello dell'applicazione e il livello del sistema principale.

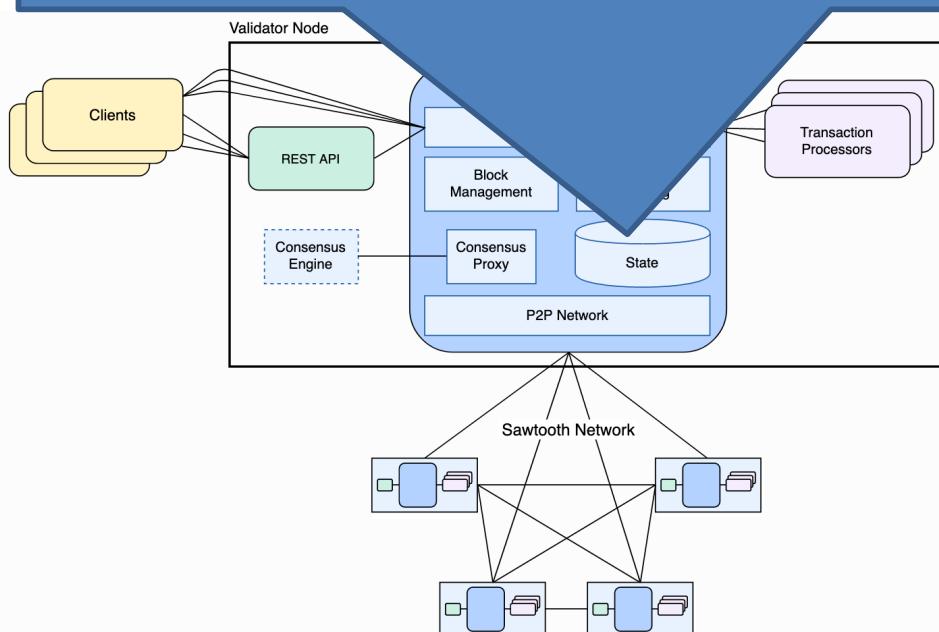


Sawtooth fornisce un'astrazione degli smart contracts che consente agli sviluppatori di applicazioni di scrivere la logica del contratto in una lingua a loro scelta, inclusi Python, Javascript, Go, C++, Java e Rust.

::: Hyperledger Sawtooth (1/4)

Hyperledger Sawtooth è una piattaforma blockchain aziendale altamente modulare che consente alle applicazioni di scegliere le transazioni. Sawtooth rappresenta lo stato per tutte le famiglie di transazioni in una singola istanza di un albero Merkle-Radix su ogni validatore, ovvero un Merkle tree indirizzabile dove gli indirizzi identificano in modo univoco i percorsi ai nodi foglia dell'albero in cui sono archiviate le informazioni.

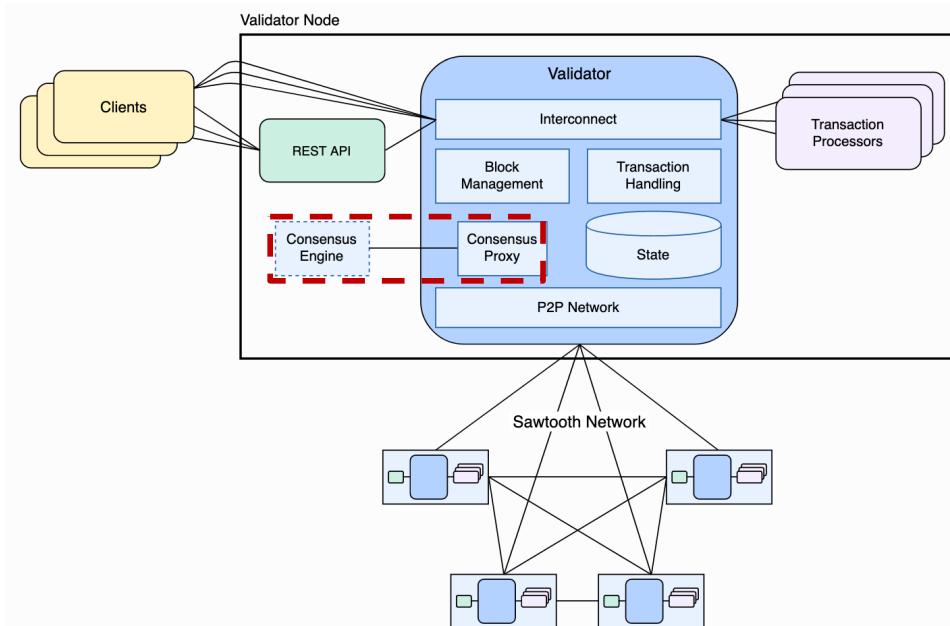
Il processo di convalida dei blocchi su ciascun validatore garantisce che le stesse transazioni risultino nelle stesse transizioni di stato e che i dati risultanti siano gli stessi per tutti i partecipanti alla rete.



Sawtooth fornisce un'astrazione degli smart contracts che consente agli sviluppatori di applicazioni di scrivere la logica del contratto in una lingua a loro scelta, inclusi Python, Javascript, Go, C++, Java e Rust.

::: Hyperledger Sawtooth (2/4)

Sawtooth astrae i concetti fondamentali del consenso e isola il consenso dalla semantica delle transazioni. L'interfaccia di consenso Sawtooth supporta il collegamento di varie implementazioni di consenso come consensus engines che interagiscono con il validatore attraverso l'API di consenso e di comunicazione P2P. Sawtooth consente di modificare il consenso dopo che la rete è stata creata ed è in esecuzione con una o due transazioni.

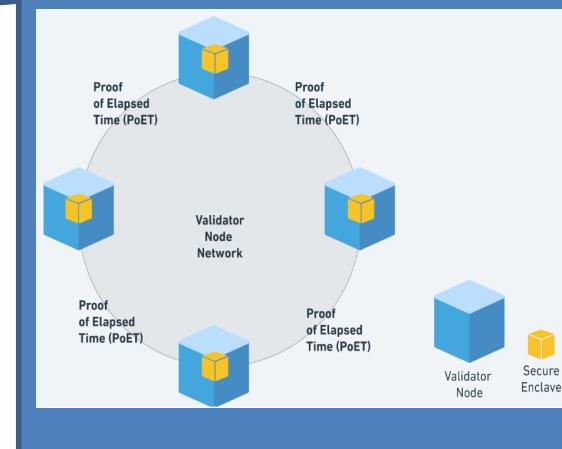
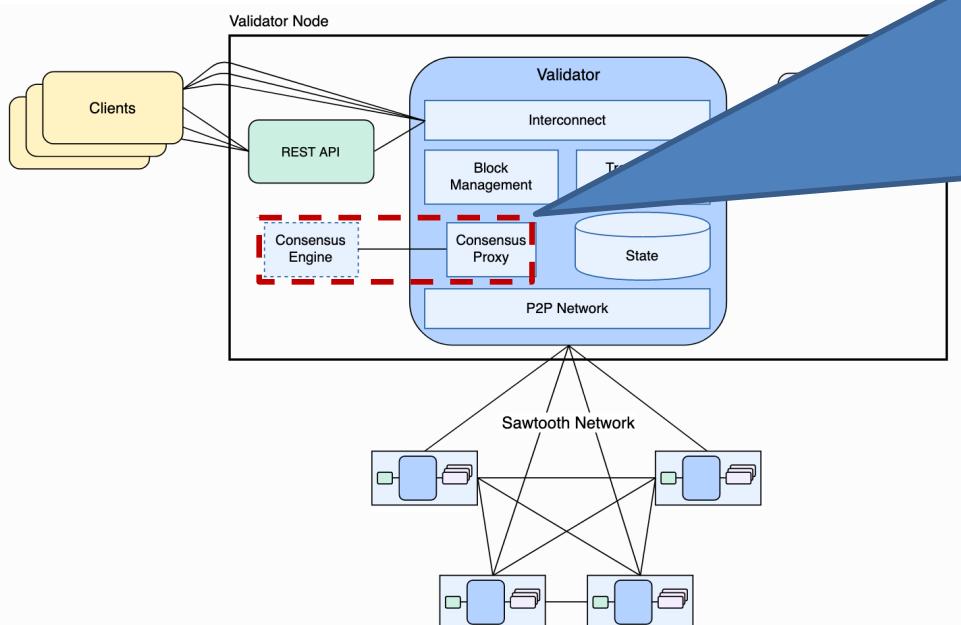


::: Hyperledger Sawtooth (2/4)

Sawtooth astrae i concetti fondamentali del consenso e isola il consenso dalla semantica delle transazioni. L'interfaccia di consenso Sawtooth supporta il collegamento di varie implementazioni di consenso come consensus engines che interagiscono con il validatore attraverso l'API di consenso e di comunicazione P2P. Sawtooth consente di modificare il consenso Proof of Elapsed Time (PoET) creata ed è in esecuzione con una

Il consenso Proof of Elapsed Time (PoET) offre una soluzione al problema dei generali bizantini che utilizza un "ambiente di esecuzione affidabile" per migliorare l'efficienza delle soluzioni attuali come Proof-of-Work.

PoET sceglie stocasticamente i peer per eseguire le richieste, e comportamenti fraudolenti sono evitati da TEE e verifica di identità.



::: Hyperledger Sawtooth (3/4)

PoET funziona essenzialmente come segue:

- Ogni validatore richiede un tempo di attesa da un'enclave (mediante l'esecuzione di una funzione attendibile).
- Il validatore con il tempo di attesa più breve per un particolare blocco di transazione viene eletto leader.
- Una funzione, come "CreateTimer", crea un timer per un blocco di transazioni che è garantito essere stato creato dall'enclave.
- Un'altra funzione, come "CheckTimer", verifica che il timer sia stato creato dall'enclave. Se il timer è scaduto, questa funzione crea un'attestazione che può essere utilizzata per verificare che il validatore abbia atteso il tempo assegnato prima di rivendicare il ruolo di leadership.

Questo algoritmo rientra nella casistica basati sulla lotteria come quello di Nakamoto.

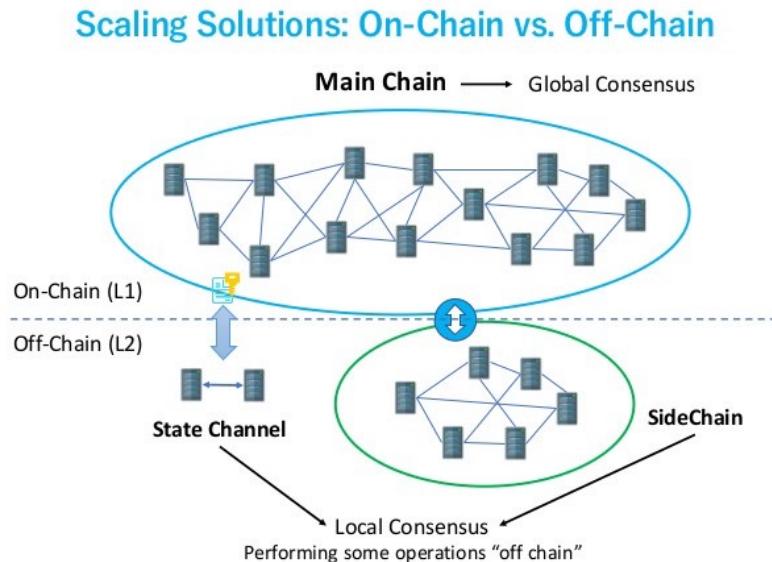
::: Hyperledger Sawtooth (4/4)

In generale, le soluzioni BFT possono resistere a comportamenti maliziosi di alcuni nodi. Le soluzioni CFT presumono che nessun nodo sia malizioso, ma può bloccarsi o scomparire dalla rete senza anomalie nel suo comportamento. Le soluzioni CFT sono generalmente meno costose e più scalabili.

- PoET con hardware SGX realizza un consenso BFT. L'enclave SGX genera in modo sicuro il valore del tempo di attesa casuale a prova di manomissione. L'enclave firma quindi un certificato con il valore del tempo di attesa. Dopo la scadenza del timer, l'attestazione SGX viene inviata agli altri nodi di rete. I nodi peer verificano la firma del tempo di attesa generata dal nodo vincente. Il nodo vincente può pubblicare il blocco proposto.
- PoET è disponibile anche senza SGX, che è simulato. Data tale simulazione, il consenso sarà solo CFT non BFT.

::: Hyperledger Avalon (1/5)

Le blockchain offrono consistenza e trust tramite una replica massiccia dei dati, ma hanno un throughput limitato e privacy e riservatezza imperfette.



On-chain sono le transazioni che consistono in una transizione di stato con consenso globale.

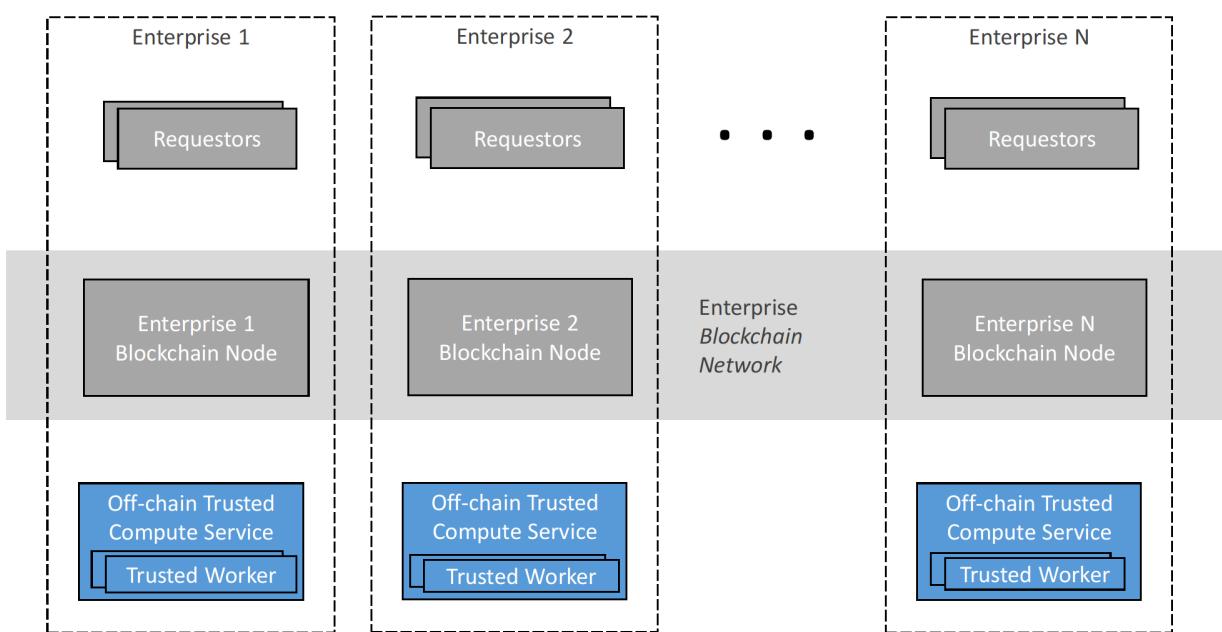
SideChain è una soluzione di pochi nodi che mantiene un consenso locale per determinate informazioni estratte dalla catena principale.

Off-chain sono elaborazioni su dati estratti dalla catena.

L'aggiunta di un'esecuzione off-chain affidabile a una blockchain è proposta come la soluzione per migliorare le prestazioni.

::: Hyperledger Avalon (2/5)

Una blockchain principale mantiene una singola istanza autorevole degli oggetti, applica le politiche di esecuzione e garantisce la verifica delle transazioni e dei risultati, mentre l'elaborazione off-chain associato consente una maggiore produttività. Si rende necessario abbinare soluzioni di trusted computing per aumentare l'integrità delle processazioni e proteggere la riservatezza dei dati.

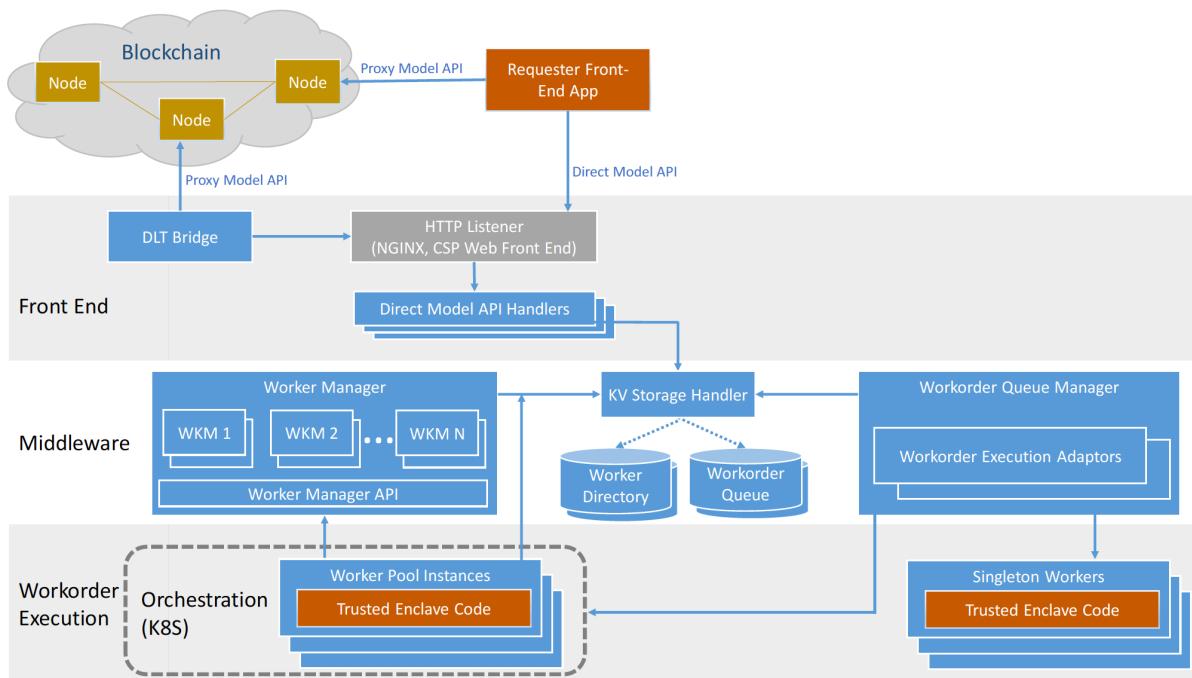


Hyperledger Avalon realizza tale soluzione.

Ogni organizzazione ha richiedenti che inviano transazioni e worker li eseguono. Le ricevute sono registrate sulla blockchain.

::: Hyperledger Avalon (3/5)

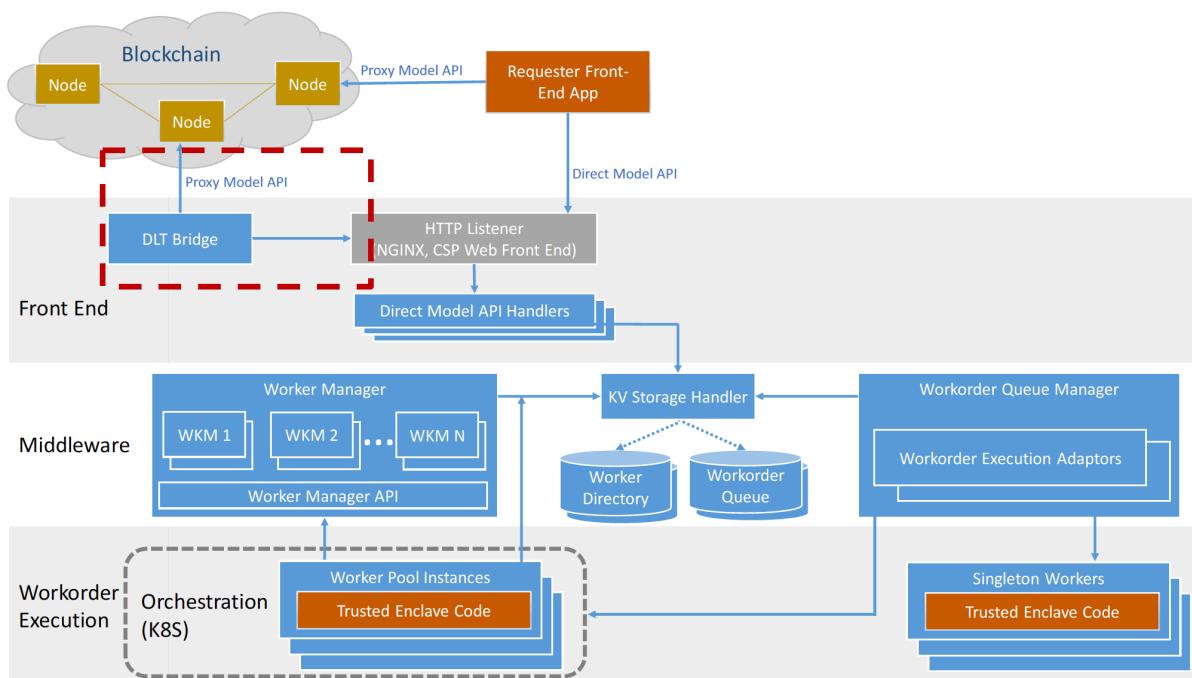
Le richieste di elaborazioni verso workers ospitati in un enclave TEE sono inviati dai richiedenti tramite l'interfaccia utente front-end o strumenti a riga di comando. Gli ordini di lavoro possono essere inviati anche da smart contract in esecuzione su DLT. Esistono due modelli di funzionamento:



::: Hyperledger Avalon (3/5)

Le richieste di elaborazioni verso workers ospitati in un enclave TEE sono inviati dai richiedenti tramite l'interfaccia utente front-end o strumenti a riga di comando. Gli ordini di lavoro possono essere inviati anche da smart contract in esecuzione su DLT. Esistono due modelli di funzionamento:

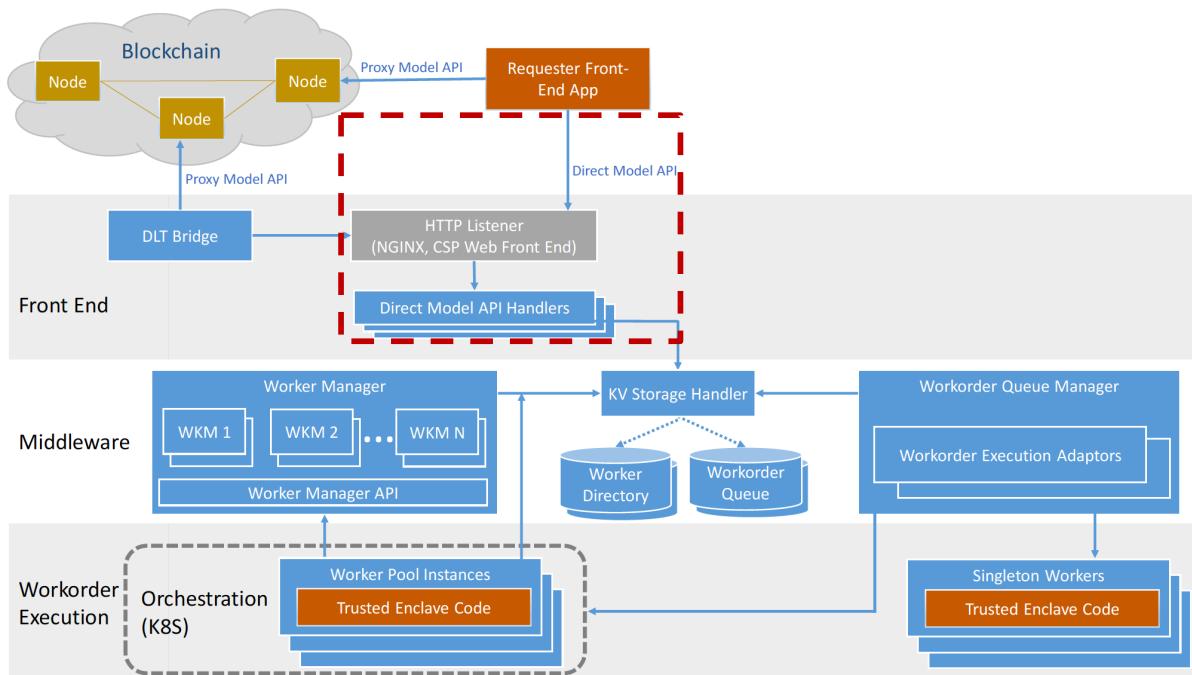
- Modello proxy per smart contract con componenti che implementano le interazioni tra DLT e TCS o connettori che astraggono le API specifiche di un DLT o un proxy



::: Hyperledger Avalon (3/5)

Le richieste di elaborazioni verso workers ospitati in un enclave TEE sono inviati dai richiedenti tramite l'interfaccia utente front-end o strumenti a riga di comando. Gli ordini di lavoro possono essere inviati anche da smart contract in esecuzione su DLT. Esistono due modelli di funzionamento:

- Modello diretto, che fornisce un'API RPC JSON.



Può essere utilizzato anche un modello ibrido che combina elementi di entrambi i modelli proxy e diretti.

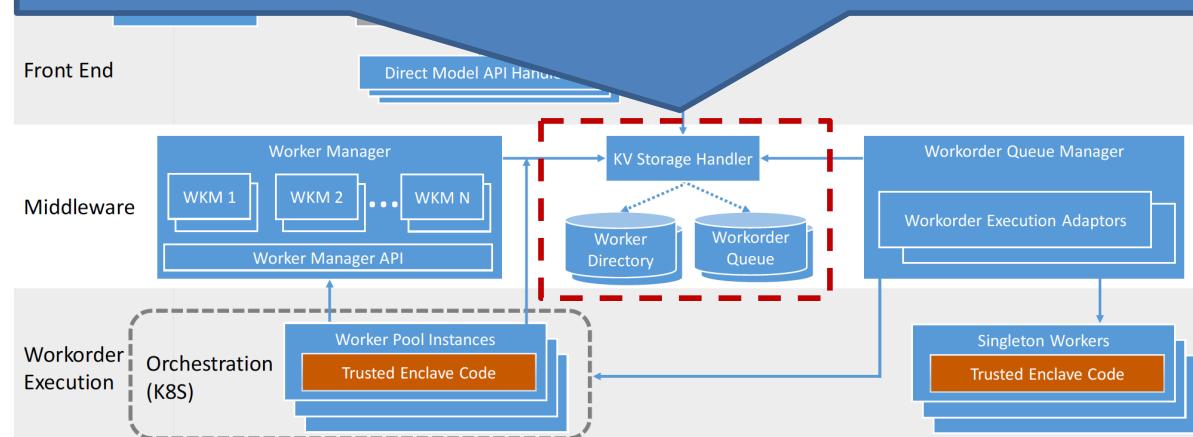
::: Hyperledger Avalon (3/5)

Le richieste di elaborazioni verso workers ospitati in un enclave TEE sono inviati dai richiedenti tramite l'interfaccia utente front-end o

Tutte le comunicazioni tra i componenti front-end e middleware vengono effettuate tramite KV Storage Manager, che mantiene:

- Una directory dei worker con tutte le informazioni sull'attestazione, sul tipo (singleton o pool) e i parametri di esecuzione (es. pod K8S, numero massimo consentito di istanze simultanee);
- Coda ordini di lavoro, che contiene richieste di ordini di lavoro, risposte e, facoltativamente, ricevute.

Il KV Storage Manager è un thin wrapper di Lightning Memory-Mapped Database (LMDB) che rappresenta una libreria software per fornire un database transazionale ad alte prestazioni sotto forma di archivio di valori-chiave.

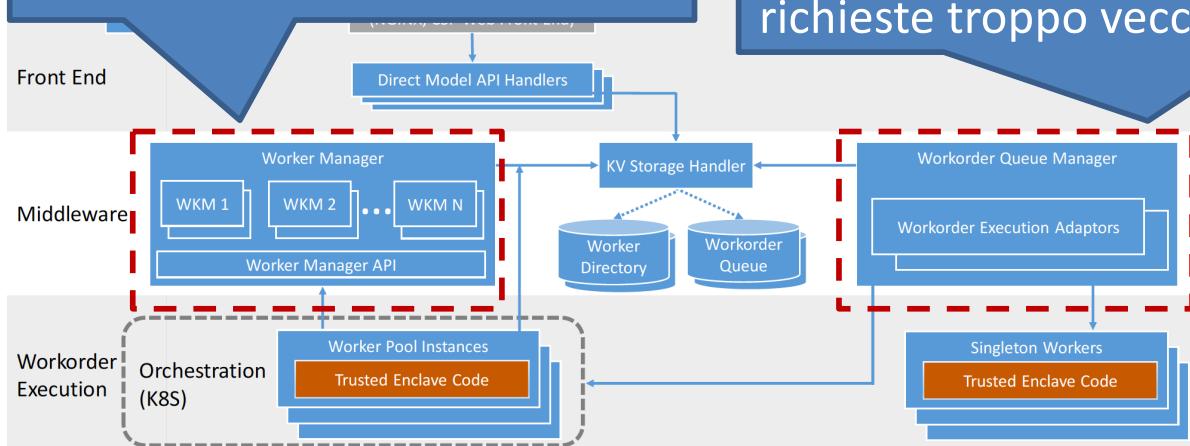


che combina elementi di entrambi i modelli proxy e diretti.

::: Hyperledger Avalon (3/5)

Le richieste di elaborazioni verso workers ospitati in un enclave TEE sono inviati dai richiedenti tramite l'interfaccia utente front-end o strumenti a riga di comando. Gli ordini di lavoro possono essere

Il Worker Manager è responsabile della creazione della chiave crittografica di ogni worker mediante Worker Key Manager (WKM) per i pool di worker, della creazione e manutenzione dei pool di worker.



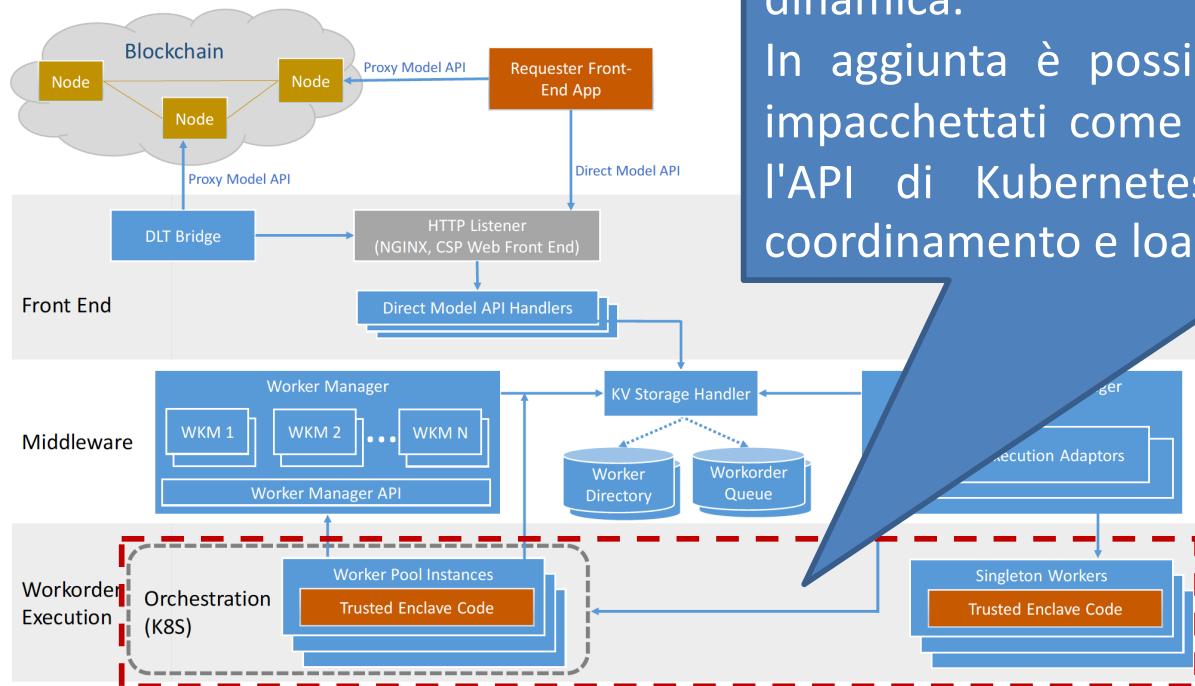
Il gestore limita il flusso di richieste per un dato worker, serializza l'esecuzione delle richieste esprimendo le dipendenze esplicitamente specificate, secondo il modello più lettura e una scrittura. Avvia l'esecuzione della richiesta tramite uno o più adattatori di esecuzione e gestisce la dimensione della coda con la rimozione di richieste troppo vecchie.

che combina elementi di entrambi i modelli proxy e diretti.

::: Hyperledger Avalon (3/5)

Le richieste di elaborazioni verso workers ospitati in un enclave TEE sono inviati dai richiedenti tramite l'interfaccia utente front-end o strumenti a riga di comando. Gli ordini di lavoro possono essere inviati anche da smart contract in esecuzione su DLT. Esistono due modelli di funzionamento:

- Modello diretto, che for

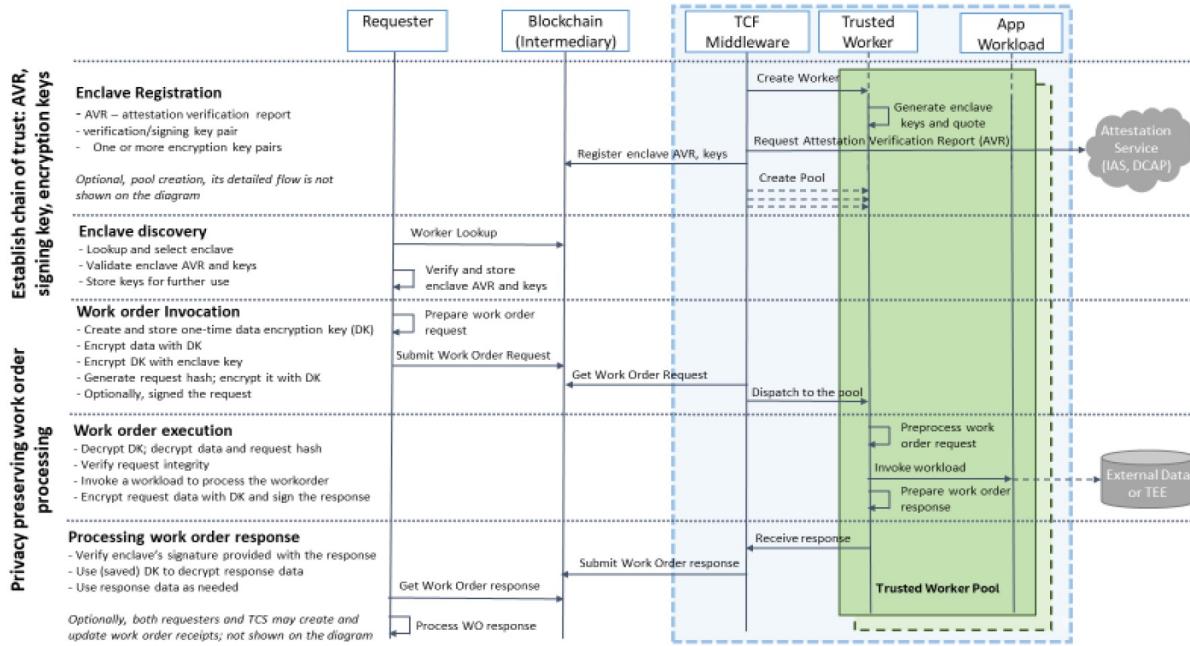


L'interazione con i worker fondamentalmente avviene con un'interazione diretta con istanze di worker allocati singolarmente o come pool, in maniera statica o dinamica.

In aggiunta è possibile prevedere che i worker siano impacchettati come pod K8S e vengono avviati tramite l'API di Kubernetes, così da avere un motore di coordinamento e loadbalancing tra di essi.

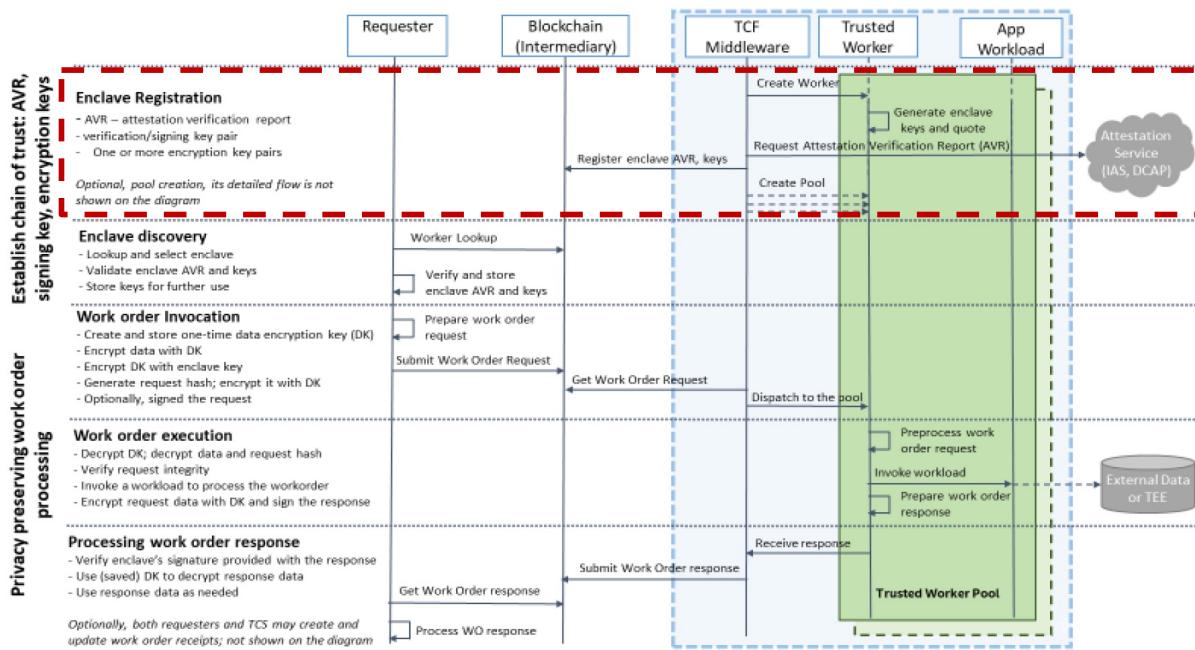
che combina elementi di entrambi i modelli proxy e diretti.

.... Hyperledger Avalon (4/5)



::: Hyperledger Avalon (4/5)

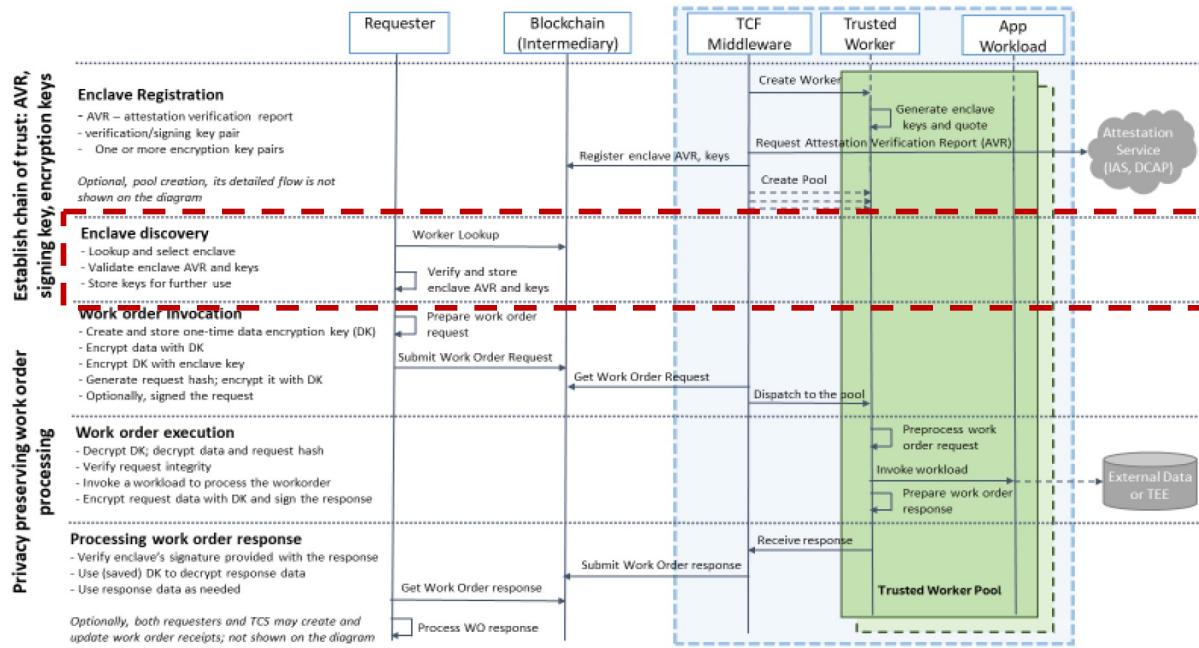
Inizialmente un nuovo worker viene istanziato in un enclave con apposito manager. Le sue informazioni di attestazione vengono generate e pubblicate sulla blockchain (o/e su un server).



::: Hyperledger Avalon (4/5)

Inizialmente un nuovo worker viene istanziato in un enclave con apposito manager. Le sue informazioni di attestazione vengono generate e pubblicate sulla blockchain (o/e su un server).

Successivamente un richiedente cerca un worker, ne verifica le informazioni di attestazione e le memorizza per un ulteriore utilizzo

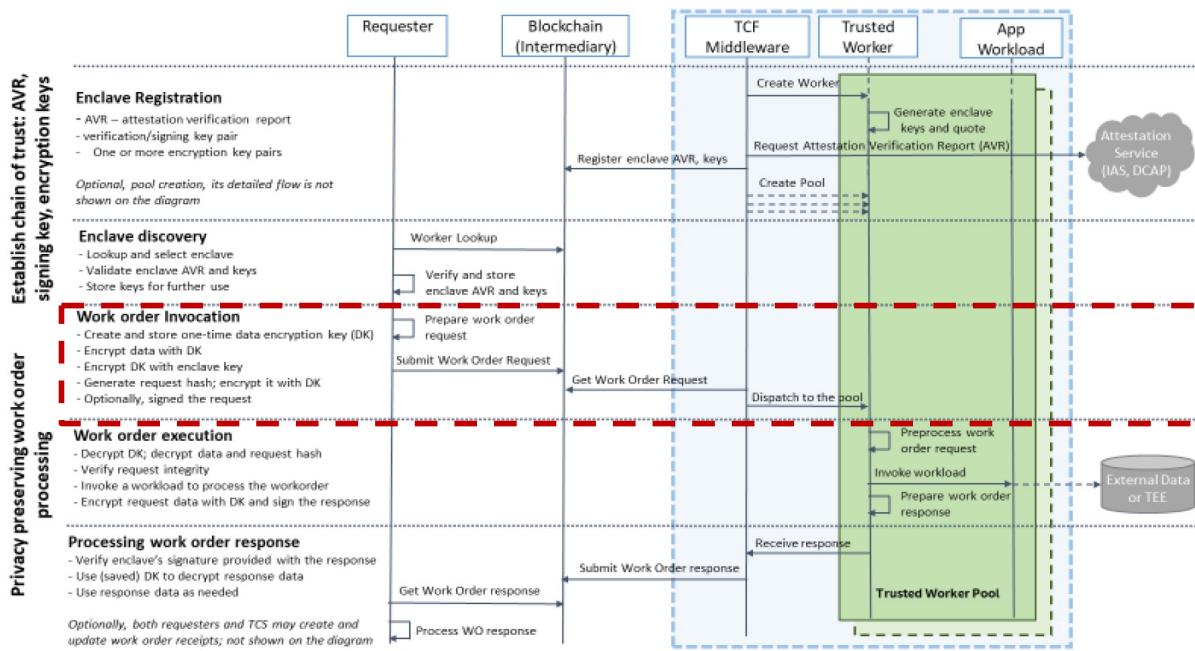


::: Hyperledger Avalon (4/5)

Inizialmente un nuovo worker viene istanziato in un enclave con apposito manager. Le sue informazioni di attestazione vengono generate e pubblicate sulla blockchain (o/e su un server).

Successivamente un richiedente cerca un worker, ne verifica le informazioni di attestazione e le memorizza per un ulteriore utilizzo

Il richiedente crea una richiesta di lavoro e lo memorizza sulla blockchain. Il manager preleva l'ordine e lo inoltra a un worker nell'enclave.

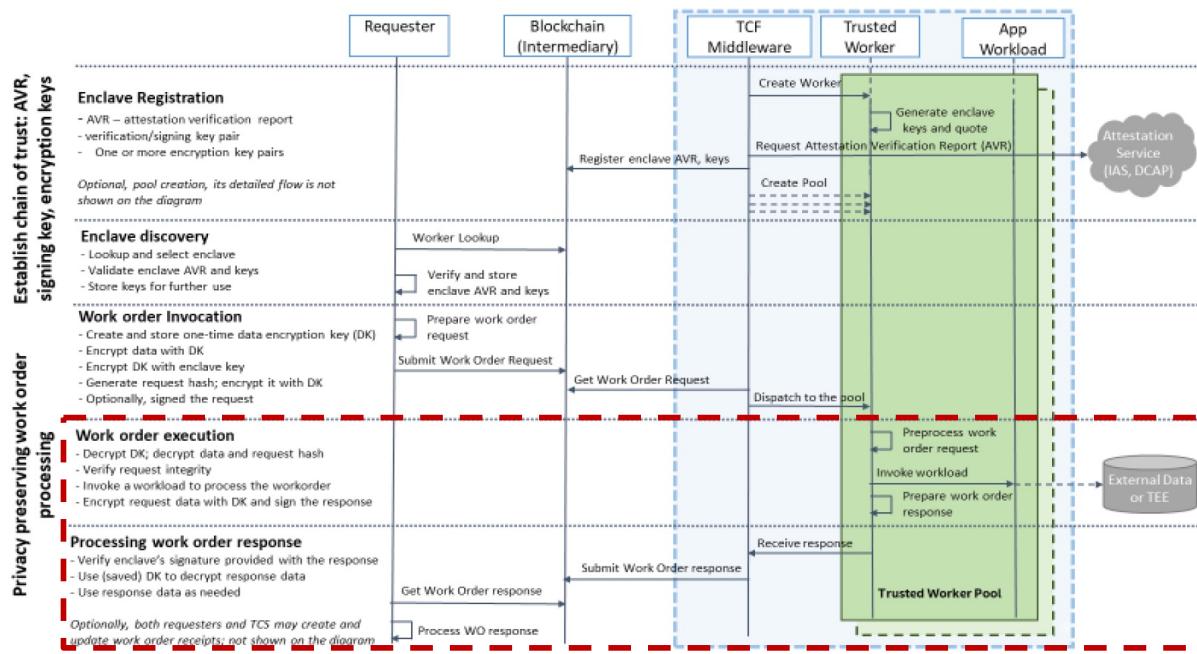


::: Hyperledger Avalon (4/5)

Inizialmente un nuovo worker viene istanziato in un enclave con apposito manager. Le sue informazioni di attestazione vengono generate e pubblicate sulla blockchain (o/e su un server).

Successivamente un richiedente cerca un worker, ne verifica le informazioni di attestazione e le memorizza per un ulteriore utilizzo

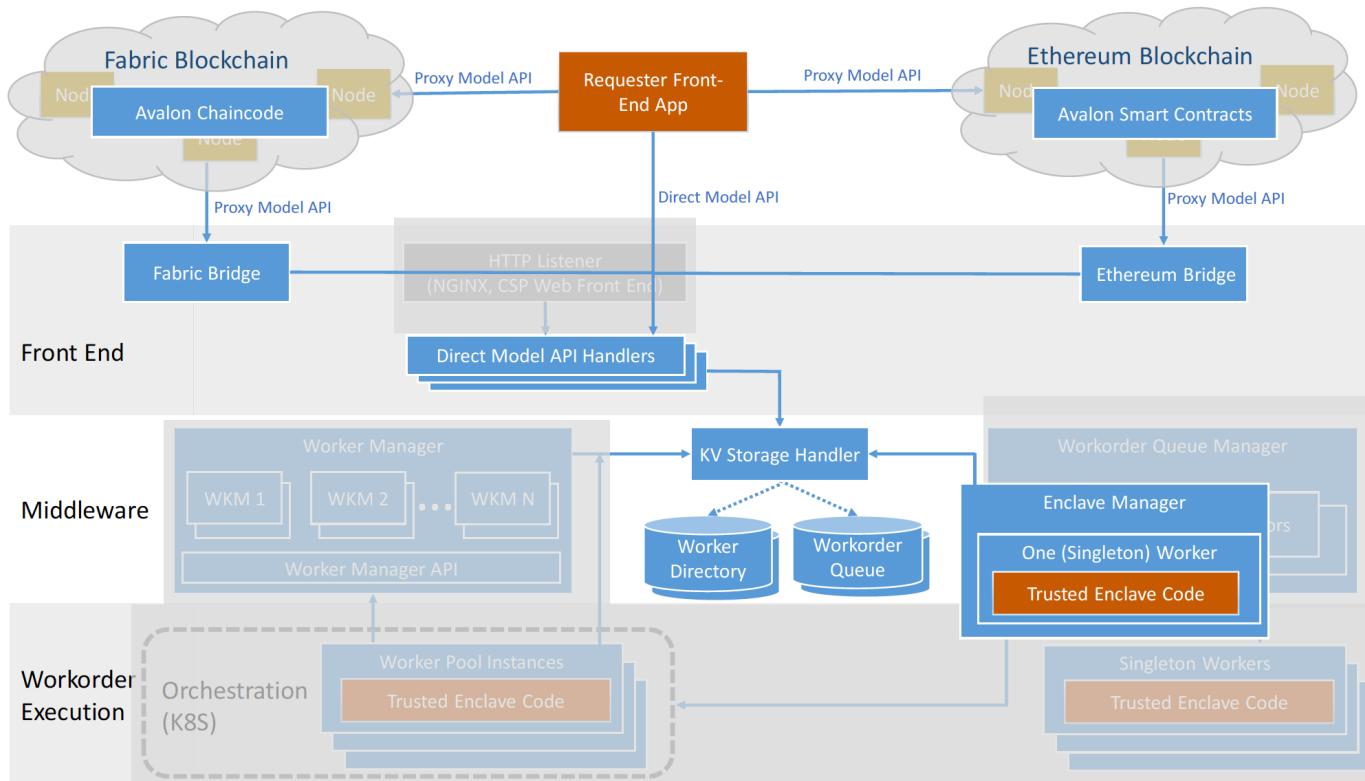
Il richiedente crea una richiesta di lavoro e lo memorizza sulla blockchain. Il manager preleva l'ordine e lo inoltra a un worker nell'enclave.



A conclusione, la risposta è posta nella blockchain e viene reperita dal richiedente.

::: Hyperledger Avalon (5/5)

L'implementazione Avalon esistente fornisce l'attestazione e la registrazione dei lavoratori, l'applicazione dell'integrità e della riservatezza degli richieste end-to-end, la loro elaborazione asincrona, il modello diretto e l'integrazione con le blockchain Fabric ed Ethereum.



Open Enclave con supporto SGX è impiegato per l'implementazione dell'enclave dove eseguire i worker.