

# Applicazione Web Energy Meters 1.0

Studio e definizione dell'architettura  
di un sistema informativo  
per l'analisi e la correlazione  
di dati estratti da database  
sui consumi energetici ad uso domestico

Attività svolta nell'ambito dell'Avviso promosso dal Ministero dell'Università e della Ricerca per la presentazione di Idee progettuali per Smart Cities and Communities and Social Innovation di cui al D.D. n. 391/Ric. del 5 luglio 2012 e ss.mm.ii..

SIN\_00968 THE LEARNING METERS NETWORK:  
workpackage formativo del SCN\_00398  
CUP J49G14000140008

Davide Costa

UNIVERSITÀ DEGLI STUDI DI PERUGIA  
Dipartimento di Ingegneria  
D.D. n. 110/2020

# Indice

<b>Indice</b>	<b>2</b>
<b>Introduzione</b>	<b>3</b>
<b>Architettura e utenti</b>	<b>4</b>
<b>Blockchain e Smart Contract</b>	<b>6</b>
Quadrans blockchain	6
Quadrans Testnet	6
Quadrans Mainnet	7
Smart Contract	7
<b>Funzionalità</b>	<b>8</b>
Funzionalità condivise	8
Funzionalità di Amministratore	9
Lambda Service	12
Certificati	13
Documento con sigillo	14
Gestione portafogli	15
Funzionalità Utente	16
<b>SQL Database</b>	<b>17</b>
Diagramma ER	17
Elenco delle tabelle	19
Lambda	19
Interfaccia Utente	19
<b>File sorgenti e cartelle</b>	<b>20</b>
<b>Hosting, Web Server e DNS</b>	<b>23</b>
Hosting	23
Web Server	23
DNS	24
Certificati SSL	24
<b>Appendice A - Esempio di PDF con Sigillo Lambda</b>	<b>25</b>
<b>Appendice B - Librerie e dipendenze</b>	<b>26</b>
<b>Appendice C - Configurazione Server Web</b>	<b>27</b>

# Introduzione

Il progetto Energy Meters si pone l'obiettivo di fornire uno strumento di controllo dei consumi energetici capillare e garantito tramite l'utilizzo combinato di dispositivi dell'Internet of Things(IoT) e della tecnologia blockchain. All'interno di questo sistema complesso ed articolato si inserisce l'**Applicazione Web Energy Meters**.

Tale applicazione ricopre diverse funzionalità fondamentali sia per quanto riguarda la gestione del sistema nel suo complesso ma soprattutto per quanto riguarda l'utilizzo del sistema da parte dell'utente finale tramite la visualizzazione dei consumi energetici come singola entità ma anche come membro facente parte di una comunità energetica.

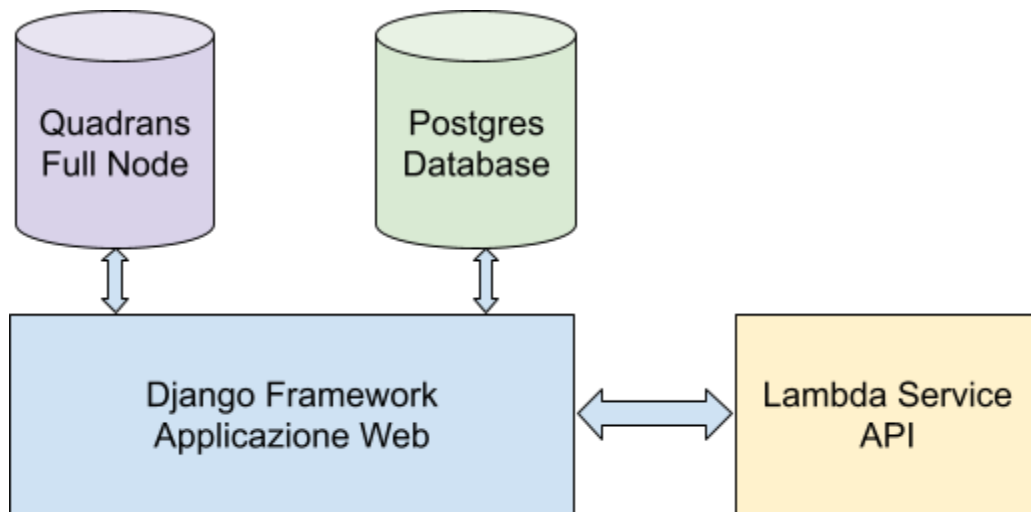
L'obiettivo dell'applicazione infatti è quello di rendere accessibile tramite una semplice interfaccia utente gli strumenti offerti dalla blockchain Quadrans ed in particolare dallo Smart Contract dei consumi energetici che però non sarà oggetto di questo documento.

Infine essa offre anche l'integrazione con Lambda Service di GT-50 di cui si approfondirà l'impiego e l'importanza nelle sezioni successive per riuscire a dimostrare la validità legale dei consumi energetici scritti immutabilmente all'interno della blockchain Quadrans.

# Architettura e utenti

L'architettura è basata su tre componenti principali open source come mostrato in **Figura 1**:

1. Quadrans Full Node versione 1.5.2
2. Database SQL Postgres versione 13.3
3. Framework Python Django 3.1.3



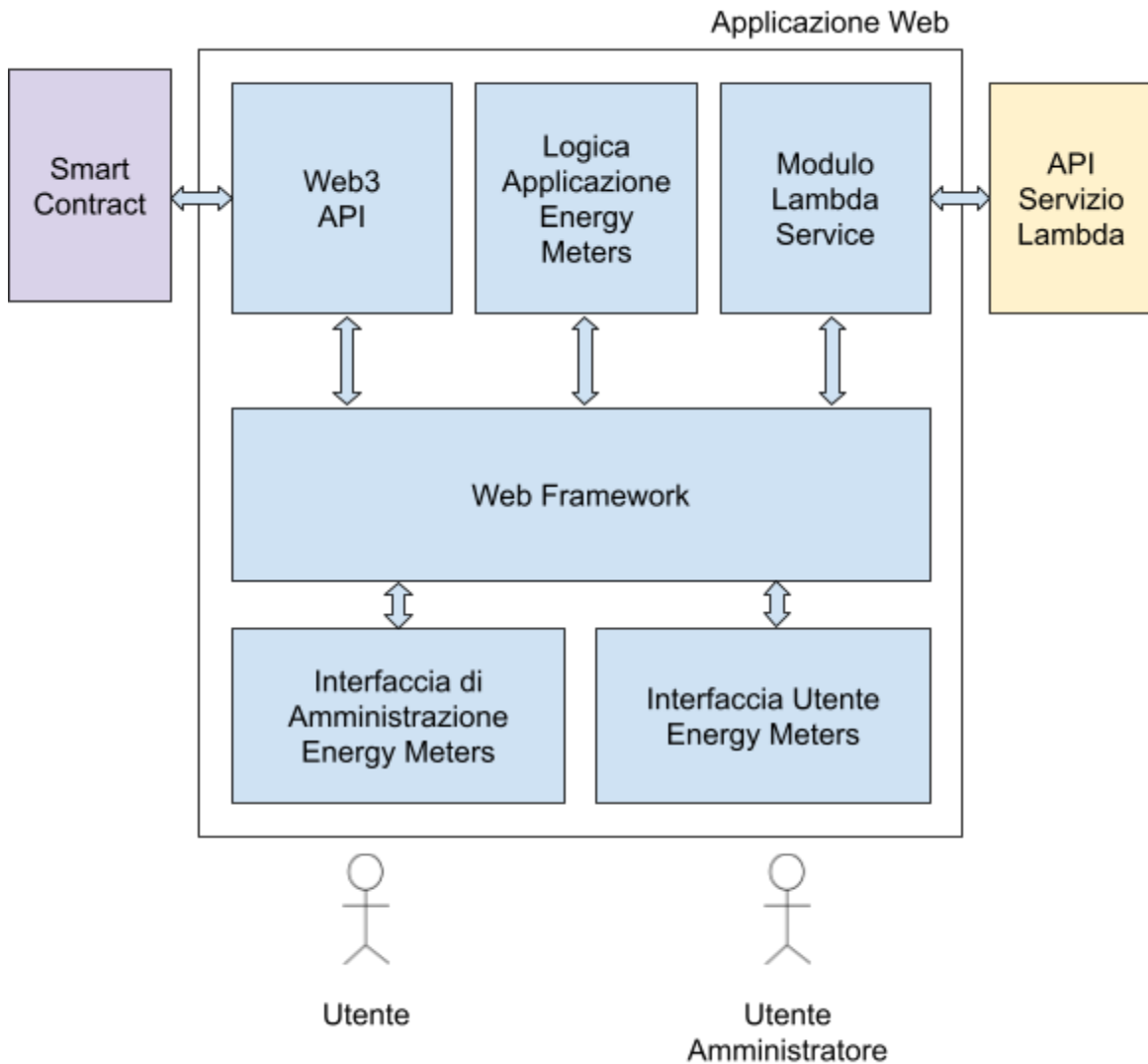
**Figura 1.** Architettura dell'Applicazione Web Energy Meters

più un servizio API esterno non open source:

4. Lambda Service API versione 1.2

integrato all'interno dell'applicazione tramite l'interfaccia API fornita. Non è oggetto di questa analisi descrivere il funzionamento di tale servizio ma ci si limiterà alla descrizione di come è stato utilizzato per raggiungere gli obiettivi del progetto.

Andando ad analizzare l'applicazione più nel dettaglio è possibile individuare le diverse interfacce utente sviluppate tramite lo schema a blocchi in **Figura 2**. Si può notare che l'interazione con la blockchain Quadrans avviene tramite l'interfacciamento con lo Smart Contract responsabile del salvataggio dei dati di consumo.

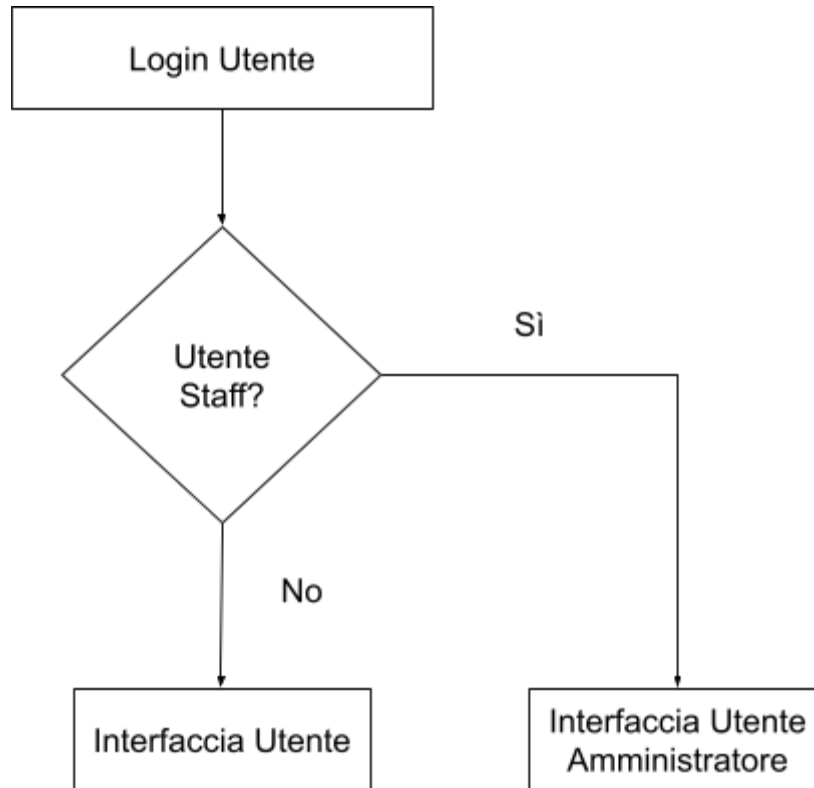


**Figura 2.** Schema a blocchi dell'Applicazione Web Energy Meters

Si configurano due diverse tipologie di utente:

1. Utente, utente utilizzatore del servizio: la registrazione avviene in autonomia da parte dell'utente stesso tramite email.
2. Utente Amministratore, utente gestore della piattaforma: la registrazione avviene tramite l'apposito pannello di amministrazione da parte di un altro utente Amministratore.

L'accesso alle due interfacce è vincolato alla tipologia di utente che effettua l'accesso secondo il diagramma di flusso in **Figura 3**.



**Figura 3.** Diagramma di flusso del reindirizzamento delle diverse tipologie di Utente

## Blockchain e Smart Contract

L'utilizzo della blockchain e degli smart contract è centrale all'interno dell'Applicazione e si vuole riportare all'interno di questo documento le informazioni di base per comprendere il funzionamento e l'integrazione della tecnologia all'interno dell'Applicazione Web.

### Quadrans blockchain

L'ecosistema Quadrans offre la rete Testnet che permette di sviluppare e testare le applicazioni che fanno uso della tecnologia prima che esse entrino in produzione sulla rete Mainnet. Verrà affrontato nei capitoli successivi come configurare l'applicazione per la rete Testnet e per la rete Mainnet. Per quanto riguarda la documentazione completa dell'ecosistema Quadrans si rimanda al sito ufficiale: <https://quadrans.io>

### Quadrans Testnet

Si riportano di seguito le informazioni principali:

<b>Network ID</b>	3
<b>Chain ID</b>	10947
<b>Explorer Address</b>	<a href="https://explorer.testnet.quadrans.io/">https://explorer.testnet.quadrans.io/</a>
<b>Tempo di blocco (medio)</b>	15 sec

## Quadrans Mainnet

Si riportano di seguito le informazioni principali:

<b>Network ID</b>	1
<b>Chain ID</b>	10947
<b>Explorer Address</b>	<a href="https://explorer.quadrans.io/">https://explorer.quadrans.io/</a>
<b>Tempo di blocco (medio)</b>	5 sec

## Smart Contract

I seguenti indirizzi dello Smart Contract sono stati forniti:

Quadrans Testnet: **0x94619d49d770d1fD68b329435A4033f5ff31f3CC**

Proprietario del wallet: **0xaF1FEB6C52c98aF20b3ad4F3C3c9620C61De15E3**

Quadrans Mainnet: **0xeFc21eBe792bEfDB807075609Eb44cE922CeBd50**

Proprietario del wallet: **0xEeB498df8bb8625E01f83F2Bc1c49180A5951b77**

Per descrivere lo Smart Contract rimangono da definire l'ABI che descrive i metodi e le variabili disponibili e gli Eventi di cui verrà approfondita la definizione e l'utilizzo nel capitolo successivo.

Gli eventi sono una parte fondamentale per meglio comprendere il funzionamento dell'applicazione:

<b>Evento</b>	<b>Parametri associati</b>	<b>Descrizione</b>
Enabled	<b>address_who</b> : indirizzo abilitato <b>address_wallet</b> : indirizzo secondario per utilizzi futuri	(*) Lanciato quando un indirizzo viene abilitato ad aggiungere o rimuovere Gateway (o Pod).

Disabled	<b>address_who</b> : indirizzo disabilitato	(*) Lanciato quando un indirizzo viene disabilitato. Automaticamente disabilita anche il parametro <b>_wallet</b> e tutti i gli indirizzi dei Gateway abilitati da questo indirizzo lanciando un evento Rejected per ognuno di essi.
Adopted	<b>address_who</b> : indirizzo abilitato <b>uint_pod</b> : identificativo del Gateway <b>address_podaddress</b> : indirizzo del Gateway	Lanciato quando viene associato l'indirizzo e il numero di un nuovo Gateway ad un indirizzo abilitato.
Rejected	<b>address_who</b> : indirizzo abilitato <b>uint_pod</b> : indirizzo abilitato <b>address_podaddress</b> : indirizzo del Gateway	Lanciato quando viene disabilitato l'indirizzo di un Gateway.
Recorded	<b>address_pod</b> : indirizzo del Gateway <b>uint_podId</b> : identificativo del Gateway <b>bytes8_sensorId</b> : identificativo del sensore <b>int32[]_data</b> : misurazione <b>uint_when</b> : timestamp <b>uint_mode</b> : modalità	Lanciato quando viene scritta una misurazione di consumo energetico.

(\*) L'applicazione prevede che l'indirizzo del proprietario dello smart contract sia l'unico che può eseguire questa operazione.

## Funzionalità

Le funzionalità vengono suddivise secondo le diverse interfacce utente all'interno delle quali sono state sviluppate. Tuttavia alcune funzionalità di base sono condivise da entrambe le interfacce.

Una o più funzionalità vengono associate all'URL corrispondente a una Vista (View) dell'Applicazione Web all'interno del paradigma MVC (Model View Controller) del framework utilizzato.

### Funzionalità condivise

Le funzionalità condivise sono quelle di base per la gestione di tutti gli account utente. Sono necessarie come in qualunque Applicazione Web per la registrazione, il login, il recupero password e altre funzioni di base.



URL	Funzione
/accounts/login/	Autorizza l'accesso
/accounts/logout/	Interrompe l'accesso
/accounts/password/change/	Cambio della password
/accounts/password/set/	Modifica della password
/accounts/inactive/	Account disattivato
/accounts/email/	Gestione email personale: cambio email, aggiunta email, rimozione email
/accounts/confirm-email/	Conferma indirizzo email
/accounts/confirm-email/(?P<key>[-:\w]+)/	Conferma indirizzo email con chiave
/accounts/password/reset/	Richiesta di cambio della password
/accounts/password/reset/done/	Cambio della password effettuato
/accounts/password/reset/key/(?P<uidb36>[0-9A-Za-z]+)(?P<key>.+)/	Richiesta di cambio della password con chiave
/accounts/password/reset/key/done/	Richiesta di cambio della password con chiave effettuato
/switch/	Effettua il controllo per indirizzare l'utente verso l'interfaccia designata

## Funzionalità di Amministratore

Le funzionalità di amministratore sono strettamente collegate alla gestione dello Smart Contract e alla gestione del database. Servono anche per avere una panoramica generale sul sistema

nel suo complesso e possono essere utilizzate da uno o più utenti amministratori anche contemporaneamente.

Per meglio comprendere il funzionamento è necessario chiarire alcuni termini utilizzati sia per quanto riguarda l'ambito di applicazione della tecnologia blockchain:

- **Portafoglio:** costituisce un file in formato JSON che contiene la chiave privata (di solito cifrata) e la relativa chiave pubblica validi per effettuare transazioni sulla rete Quadrans.
- **Indirizzo:** è una stringa alfanumerica che permette di identificare un portafoglio; viene derivata dalla chiave pubblica contenuta nel portafoglio.
- **Proprietario dello Smart Contract:** Portafoglio che ha creato lo Smart Contract
- **Evento:** la blockchain è una lista di blocchi che sono fondamentalmente liste di transazioni. Ogni transazione ha una ricevuta allegata che contiene zero o più voci di registro. Le voci di registro rappresentano il risultato degli eventi che sono stati lanciati da uno Smart Contract.
- **Quadrans testnet:** rete blockchain Quadrans utilizzabile per lo sviluppo e test di applicazioni.
- **Quadrans mainnet:** rete blockchain Quadrans di produzione.
- **Quadrans coin:** sono asset crittografici usati per registrare informazioni sulla blockchain Quadrans (<https://quadrans.io/ecosystem.php>).

sia per l'ambito specifico dell'applicazione IoT:

- **Gateway:** Un gateway dell'internet delle cose (IoT) è un dispositivo fisico o un programma software che serve come punto di connessione tra il cloud e i controller, i sensori e i dispositivi intelligenti. Tutti i dati che si muovono tra i dispositivi IoT e il cloud passano attraverso un gateway IoT, che può essere un dispositivo hardware dedicato o un programma software. Un gateway IoT può anche essere definito un gateway intelligente.

che per l'ambito informatico in generale:

- **Cronjob:** L'utilità software cron, conosciuta anche come cron job, è un job scheduler basato sul tempo nei sistemi operativi Unix-like.
- **Thread:** In informatica, un thread di esecuzione è la più piccola sequenza di istruzioni programmate che può essere gestita indipendentemente da uno scheduler, che è tipicamente una parte del sistema operativo.
- **Screen:** Il comando screen in Linux fornisce la possibilità di lanciare e utilizzare più sessioni di shell da una singola sessione ssh. Quando un processo viene avviato con 'screen', il processo può essere staccato dalla sessione e poi può riattaccare la sessione in un momento successivo. Quando la sessione è staccata, il processo che è stato originariamente avviato dallo schermo è ancora in esecuzione e gestito dallo schermo stesso. Il processo può poi riattaccare la sessione in un momento successivo, e i terminali sono ancora lì, nel modo in cui è stato lasciato.

URL	Method	Funzione
/unlock/	GET	Fornisce il form per effettuare le operazioni descritte nel metodo POST
	POST	Provvede allo sblocco del portafoglio del proprietario dello Smart Contract. Prende come parametro la <b>password</b> con la quale è stato cifrato il portafoglio
/lock/	POST	Provvede al blocco del portafoglio del proprietario dello Smart Contract
/contracts/dashboard/	GET	Fornisce una panoramica su: <ul style="list-style-type: none"> <li>• Numero di indirizzi attivi</li> <li>• Numero di Gateway attivi</li> <li>• Numero di consumi registrati (eventi <b>Recorded</b>)</li> <li>• Indirizzo del proprietario dello Smart Contract</li> <li>• Indirizzo dello Smart Contract in uso</li> <li>• Rete Quadrans in uso (Tetsnet, Mainnet)</li> </ul>
/contracts/gateway-events/	GET	Fornisce l'elenco degli eventi specificati nello Smart Contract in ordine cronologico suddivisi per nuovi indirizzo: <ul style="list-style-type: none"> <li>• <b>Enabled</b>: abilitazione di un nuovo indirizzo ad aggiungere o rimuovere nuovi gateway</li> <li>• <b>Disabled</b>: disabilitazione di un nuovo indirizzo ad aggiungere o rimuovere nuovi gateway</li> </ul> e nuovi gateway: <ul style="list-style-type: none"> <li>• <b>Adopted</b>: aggiunta di un nuovo gateway</li> <li>• <b>Rejected</b>: rimozione di un gateway</li> </ul>
/contracts/record-events/	GET	Fornisce l'elenco di tutti gli eventi di tipo <b>Recorded</b>
/contracts/gateway-management/	GET	Fornisce: <ul style="list-style-type: none"> <li>• l'elenco degli indirizzi abilitati ad aggiungere o rimuovere un gateway</li> <li>• l'elenco dei gateway abilitati, l'importo in Quadrans Coin relativo all'indirizzo e il numero identificativo del gateway</li> </ul>

/contracts/owner/	GET	Fornisce il form per effettuare le operazioni descritte nel metodo POST
	POST	Aggiungi Indirizzo: <ul style="list-style-type: none"> <li>• Abilita un indirizzo che a sua volta può aggiungere o rimuovere nuovi gateway</li> </ul> Disabilita Indirizzo: <ul style="list-style-type: none"> <li>• Disabilita un indirizzo</li> </ul>
/contracts/pod/	GET	Fornisce il form per effettuare le operazioni descritte nel metodo POST
	POST	Aggiungi Gateway: <ul style="list-style-type: none"> <li>• Autorizza un gateway ad effettuare transazioni di scrittura dei consumi energetici</li> </ul> Rimuovi Gateway: <ul style="list-style-type: none"> <li>• Rimuove l'abilitazione del gateway ad effettuare transazioni di scrittura dei consumi energetici</li> </ul>
/lambda/seal/	GET	Fornisce l'elenco di tutti i documenti PDF originali e il corrispondente documento PDF firmato digitalmente tramite il sigillo offerto dal servizio Lambda
	POST	Permette di impostare la creazione automatica dei sigilli Lambda come spiegato nell'apposita sezione
/admin/	GET	Permette di accedere al pannello di gestione del database offerto dal framework Python Django: <a href="https://docs.djangoproject.com/en/3.1/ref/contrib/admin/">https://docs.djangoproject.com/en/3.1/ref/contrib/admin/</a>

## Lambda Service

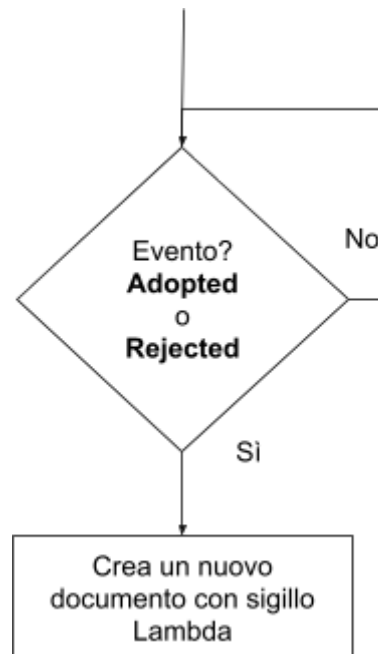
Non è obiettivo di questo documento descrivere il Lambda Service di GT-50 e si rimanda alla pagina ufficiale <https://www.gt50.org/products/IT/lambdaservice/> e alla presentazione [https://www.gt50.org/products/IT/lambdaservice/PRE\\_Lambda\\_Services.pdf](https://www.gt50.org/products/IT/lambdaservice/PRE_Lambda_Services.pdf) per ulteriori approfondimenti.

Si vuole tuttavia descrivere l'utilizzo e le motivazioni dietro la metodologia utilizzata per sfruttare al meglio questo servizio per gli scopi del progetto. La modalità di utilizzo scelta per il servizio Lambda ha l'unico obiettivo di fornire valore legale ai dati di consumo domestico misurati dal sistema tramite Gateway IOT, sensori o qualunque altra metodologia prevista dal progetto.

Per raggiungere questo obiettivo si è deciso di abbinare il sigillo Lambda ad un documento pdf contenente tutti gli indirizzi corrispondenti ai portafogli abilitati a lanciare eventi di tipo **Recorded** tramite lo smart contract. Sebbene non ci sia una garanzia che questi dati acquistino valore

legale è possibile assumere che si possa creare un collegamento dimostrabile crittograficamente tra il documento firmato digitalmente e i portafogli che firmano le transazioni.

Per ottenere questo risultato è stato creato uno script in grado di “ascoltare” qualunque cambiamento relativo alla lista degli indirizzi abilitati e automaticamente generare un nuovo documento tramite il servizio Lambda.



Dal punto di vista applicativo questa logica di monitoraggio in tempo reale degli eventi lanciati dallo smart contract offre alcuni vantaggi rispetto ad un’implementazione basata su un cronjob. Nella pratica lo script crea due Thread differenti che rimangono “in ascolto” su ciascun dei seguenti eventi:

- Adopted
- Rejected

Nel momento in cui uno di questi due eventi viene recepito dallo script viene creato un nuovo documento pdf e viene richiesto il sigillo tramite il servizio Lambda.

Lo script viene lanciato tramite il comando screen in modo tale da mantenere l’esecuzione attiva anche a seguito della disconnessione dell’utente che lo ha lanciato.

Esempio:

```
# screen  
# source /home/projects/venv-energy-chain/bin/activate  
# cd /home/projects/energy-chain/energy_meteters
```

```
# python manage.py runscript lambda_async
# CTRL+A+D
```

## Certificati GT-50

Sono stati salvati i certificati necessari all'utilizzo delle API del servizio lambda sul disco della VPS messa a disposizione e di cui si approfondiranno i dettagli in seguito. L'elenco dei certificati forniti è:

- **CA.GT50-201911.crt** chiave pubblica del certificato della CA di GT-50 che ha generato la chiave pubblica del webserver dell'appliance, in formato X509.
- **UniPerugiaCli1.8200B8.p12** chiave PRIVATA del certificato di autenticazione, in formato PKCS12 e JKS, la cui password di protezione viene inviata tramite SMS a un numero di fiducia che dovete comunicare.
- **ca\_gt50.crt** chiave PUBBLICA del certificato della nostra CA che ha generato la chiave privata precedete, in formato X509
- **Pes-app.300004.crt** chiave PUBBLICA del certificato del webserver dell'appliance, in formato X509 e JKS

Inoltre, per permettere il corretto funzionamento della libreria scelta per la comunicazione HTTP (<https://httplib2.readthedocs.io>) tra l'Applicazione Web e le API del servizio Lambda si reso necessario generare i seguenti file:

- **UniPerugiaCli1.key.pem**

### Tramite il comando:

```
$ openssl pkcs12 -in UniPerugiaCli1.8200B8.p12 -out UniPerugiaCli1.key.pem -nocerts -nodes
```

- **UniPerugiaCli1.crt.pem**

### Tramite il comando:

```
$ openssl pkcs12 -in UniPerugiaCli1.8200B8.p12 -out UniPerugiaCli1.crt.pem -clcerts -nokeys
```

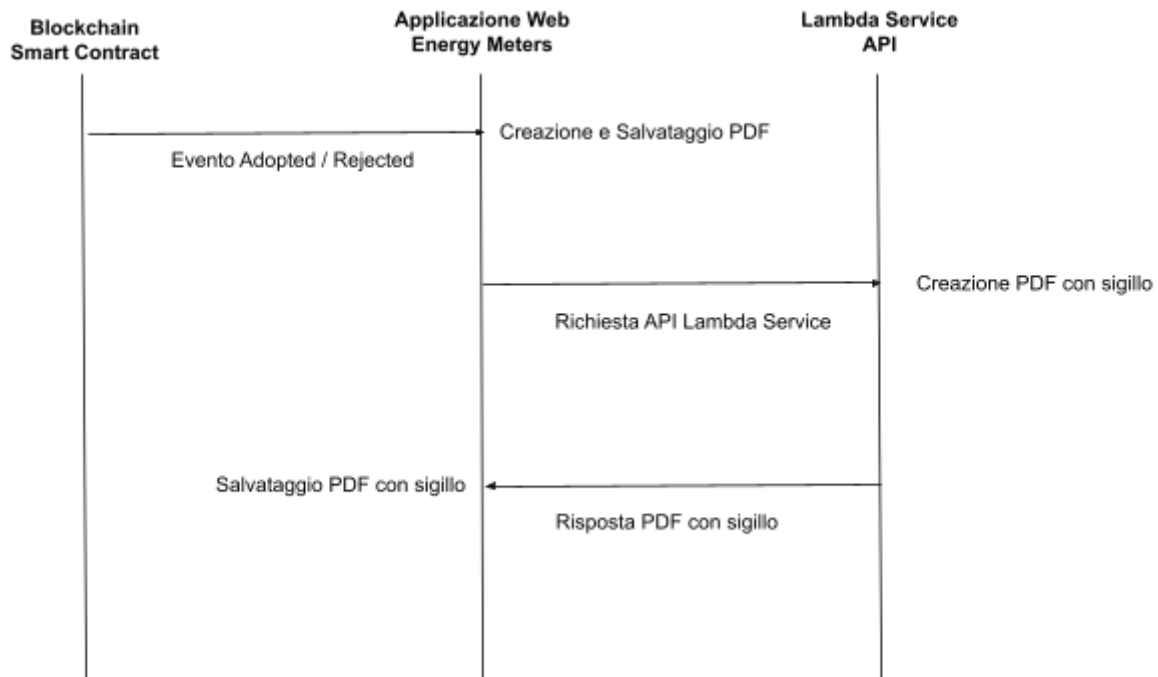
## Documento con sigillo

Il documento Lambda con sigillo nasce come documento PDF creato dinamicamente (Appendice B) in carta intestata del progetto e contenente le seguenti informazioni:

- Indirizzo dello Smart Contract a cui si fa riferimento

- Lista di indirizzi abilitati alla creazione di eventi di tipo Recorded

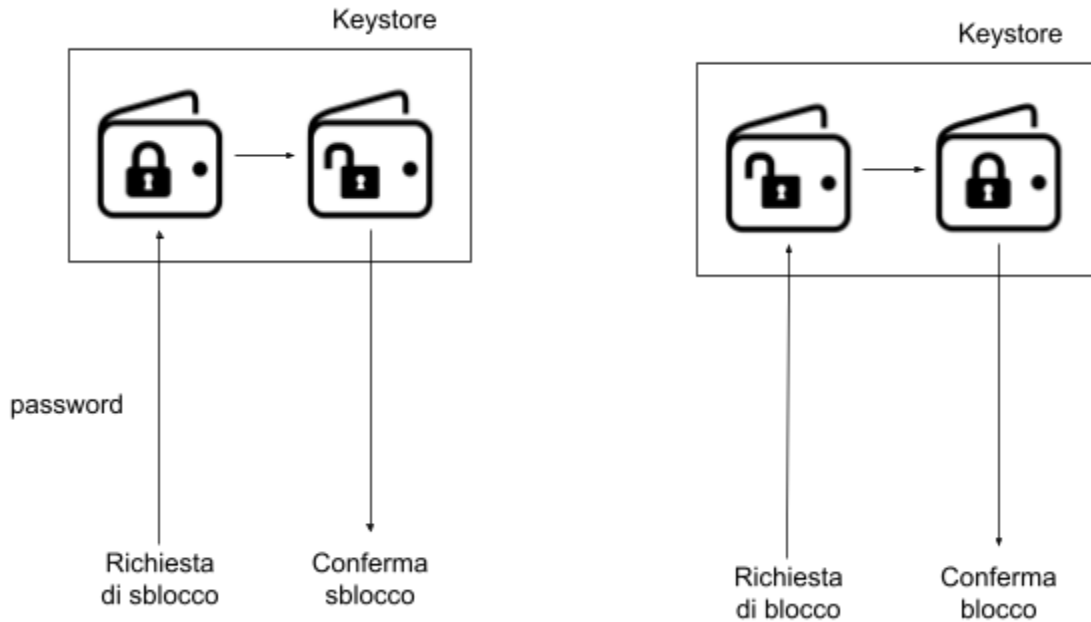
A seguito della creazione del documento viene effettuata la richiesta di creazione del documento con l'aggiunta del sigillo Lambda come mostrato in **Figura 4**.



**Figura 4.** Flusso di creazione dei documenti con sigillo Lambda

## Gestione portafogli

Il portafoglio del proprietario dello Smart Contract viene gestito dal Quadrans Full Node. Per permettere l'utilizzo della sezione di amministrazione è necessario decifrare tale portafoglio tramite una richiesta di sblocco tramite password; per cifrare nuovamente il portafoglio è sufficiente effettuare una richiesta di blocco come mostrato in **Figura 5**. Se l'utente amministratore non dovesse effettuare la richiesta di blocco il portafoglio si bloccherà automaticamente dopo un tempo configurato nel file di impostazioni che verrà affrontato nelle sezioni successive.



**Figura 5.** Diagramma di flusso del blocco e sblocco del portafoglio del proprietario dello smart contract

Questa metodologia presenta alcune criticità dal punto di vista della sicurezza ma permette in maniera semplice di effettuare le transazioni necessarie al corretto funzionamento del sistema. Un possibile miglioramento della gestione potrebbe essere l'utilizzo di software quali Metamask o simili sempre compatibili con la blockchain Quadrans.

## Funzionalità Utente

Le funzionalità utente sono molto semplici e rispecchiano quelle che sono le indicazioni fornite per l'esperienza utente. Per meglio comprendere tali funzionalità si rende necessario spiegare la terminologia:

- **Sensore:** dispositivo che effettivamente effettua la misurazione dei consumi relativi ad un elettrodomestico o di qualunque dispositivo che consuma energia
- **Etichetta:** corrisponde ad una tipologia predefinita di elettrodomestico o dispositivo che consuma energia, ad esempio: frigorifero.

URL	Method	Funzione
/user_interface/profile/	GET	Permette la visualizzazione del profilo utente e ne fornisce il form per la modifica o creazione



	POST	Permette: <ul style="list-style-type: none"> <li>● la creazione del profilo utente</li> <li>● la modifica del profilo utente</li> <li>● l'associazione dell'indirizzo gateway con l'account utente</li> </ul>
/user_interface/dashboard/	GET	Fornisce una panoramica su: <ul style="list-style-type: none"> <li>● i sensori attivi associati al gateway impostato nel profilo utente</li> <li>● I consumi di ciascun <b>Sensore</b></li> <li>● <b>Etichette</b> attive associate a ciascun <b>Sensore</b></li> </ul>
	POST	Permette di: <ul style="list-style-type: none"> <li>● Associare il codice di un <b>Sensore</b> con un'<b>Etichetta</b></li> <li>● Modificare l'<b>Etichetta</b> associata ad un <b>Sensore</b></li> </ul>
/user_interface/history/	GET	Permette di visualizzare lo storico dei consumi energetici di ogni <b>Sensore</b> , le <b>Etichette</b> associate al <b>Sensore</b> e gli eventuali cambi di <b>Etichette</b> con relativa data di modifica
/user_interface/compare/	GET	Permette di confrontare i consumi energetici associati alle <b>Etichette</b> dell'utente con quelli della comunità energetica fornendo un valore percentile rispetto alla comunità
	POST	Permette di generare il confronto tra le <b>Etichette</b> dell'utente e quelle della comunità energetica

I valori misurati dai sensori sono espressi in mW, la formula di conversione per ottenere i kW è banalmente lettura/10<sup>6</sup>, mentre per ottenere i kW/h viene usata la seguente :

$$E \text{ (kWh)} = (0.5 * (P1 + P0) / 10^6) * ((t1 - t0) / 3600)$$

dove: (P0, t0) è la rilevazione meno recente

Con P0 espressa in mW e t0 (timestamp in secondi) - (P1, t1) è la rilevazione "più recente" con P1 espressa in mW e t1 (timestamp in secondi)

# SQL Database

## Diagramma ER

Per permettere il corretto funzionamento del sistema si è reso necessario l'utilizzo di un database relazionale. Si è cercato di ridurre al minimo indispensabile l'utilizzo del database centralizzato privilegiando quanto più possibile l'utilizzo dei dati on-chain disponibili tramite lo smart contract.

In **Figura 6** viene mostrato il diagramma ER del database.



## Elenco delle tabelle

L'elenco delle tabelle si limita a quelle necessarie all'Applicazione Web, suddiviso per moduli, in quanto il framework scelto e le eventuali librerie aggiuntive che necessitano dell'utilizzo del database sono ben documentate all'interno della documentazione ufficiale:

- Django: <https://docs.djangoproject.com/en/3.1/>
- Django Allauth: <https://django-allauth.readthedocs.io/en/latest/>

### Lambda

<b>LambdaSettings</b> <SingletonModel>		
<b>id</b>	<b>BigAutoField</b>	<b>Chiave Primaria</b>
enable_contract_listening	BooleanField	Abilita la creazione automatica dei documenti PDF tramite Lambda service

<b>PdfSeal</b>		
<b>id</b>	<b>BigAutoField</b>	<b>Chiave Primaria</b>
creation_date	DateTimeField	Timestamp
file_name	CharField	Nome file PDF
file_name_seal	CharField	Nome file PDF con sigillo Lambda
file_pdf	FileField	File PDF
file_pdf_seal	FileField	File PDF con sigillo Lambda

### Interfaccia Utente

Profile		
id	BigAutoField	Chiave Primaria
user	OneToOneField (id)	Chiave Esterna
cap	IntegerField	CAP dell'utente
picture	ImageField	Immagine profilo utente
wallet	CharField	Indirizzo del Gateway dell'utente
terms_and_conditions	BooleanField	Accettazione termini e condizioni

Tag		
id	BigAutoField	Chiave Primaria
user	ForeignKey (id)	Chiave Esterna
from_block	IntegerField	Blocco della blockchain Quadrans di inizio validità del Tag
name	CharField	Nome del Tag
pod_id	CharField	Identificativo del Gateway
sensor_id	CharField	Identificativo del Sensore
to_block	IntegerField	Blocco della blockchain Quadrans di fine validità del Tag

## File sorgenti e cartelle

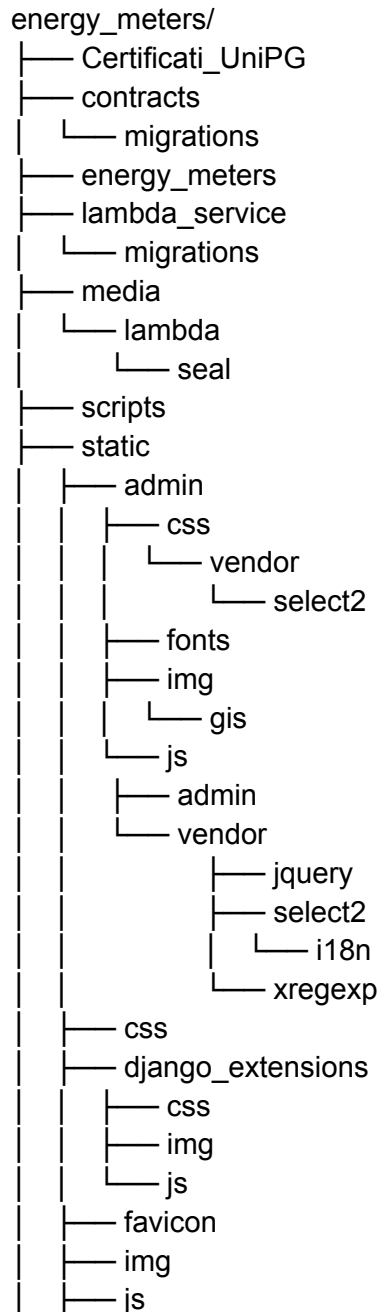
Tutti i file fatta eccezione per i Certificati forniti da GT-50 sono disponibili all'interno del repository dedicato e che costituisce il riferimento per eventuali correzioni e/o risoluzione di anomalie: [https://github.com/davidecosta/energy\\_meters](https://github.com/davidecosta/energy_meters)

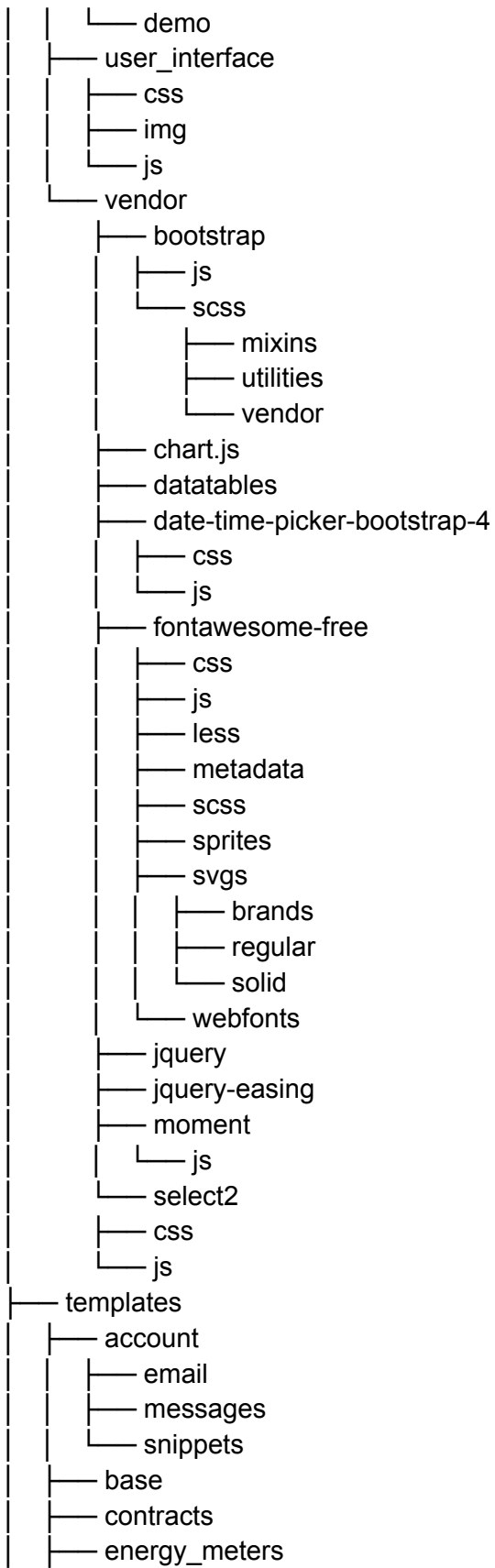
Una copia di questo repository è attualmente installata sulla Macchina Virtuale in esercizio fornita e di cui si approfondirà la configurazione nel capitolo successivo. La struttura dei file è suddivisa in moduli. Ciascun modulo corrisponde ad un set di funzionalità. I moduli applicativi sono:

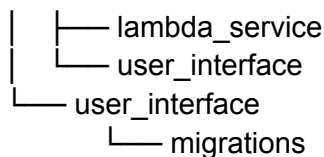
- **contracts**: responsabile per la gestione dello smart contract
- **lambda\_service**: responsabile per la gestione dell'integrazione col Servizio Lambda
- **user\_interface**: responsabile per la gestione della parte utente dell'applicazione

Si faccia riferimento alla documentazione del Framework Django per ulteriori informazioni su

- Templates: <https://docs.djangoproject.com/en/3.1/topics/templates/>
- File statici: <https://docs.djangoproject.com/en/3.1/howto/static-files/>
- File media: <https://docs.djangoproject.com/en/3.1/topics/files/>







## Hosting, Web Server e DNS

### Hosting

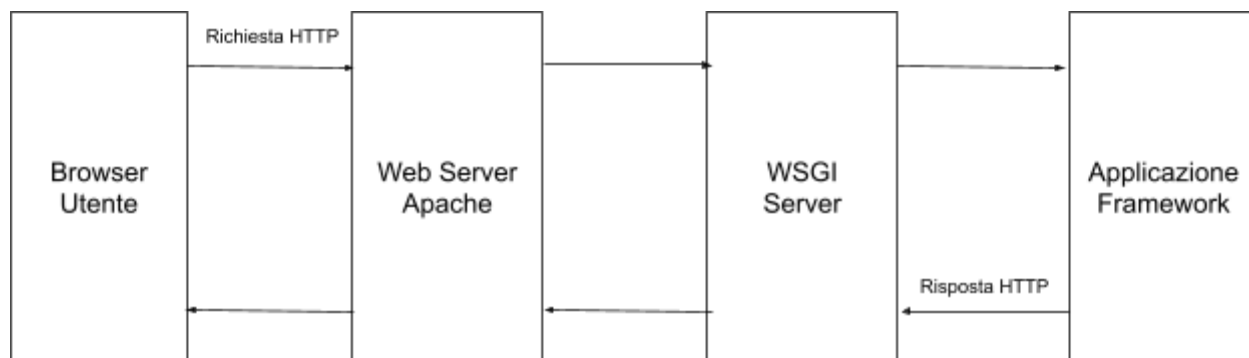
L'applicazione Web è attualmente in funzione sulla VPS (Virtual Private Server) messa a disposizione. Di seguito le informazioni per l'utilizzo del sistema

<b>Provider</b>	https://www.ionos.it
<b>Indirizzo IPv4</b>	217.160.191.76
<b>vCPU</b>	4
<b>RAM</b>	8GB
<b>Spazio su Disco</b>	160GB
<b>Tipo di Disco</b>	SSD
<b>Sistema Operativo</b>	Ubuntu 20.04.2 LTS
<b>Kernel Linux</b>	GNU/Linux 5.4.0-77-generic x86_64
<b>Velocità di trasferimento di picco</b>	400 Mbps

### Web Server

Il web server scelto per l'applicazione è Apache (apache2) con modulo WSGI (mod\_wsgi). In **Figura 7** è mostrato uno schema che rappresenta schematicamente l'architettura del Server. Un esempio di configurazione del server è fornito in Appendice C





**Figura 7.** Schema a blocchi del server web Energy Meters

## DNS

Il Domain Name Server al quale risponde l'Applicazione web è: **dapp.energychain.it**

Allo stesso indirizzo IP è configurato anche l'URL dell'applicazione di test:

**test-dapp.energychain.it**

usata per permettere i primi test operativi condotti negli ultimi mesi dello sviluppo.

Entrambi sono stati configurati su <https://www.register.it>

Tuttavia solo una delle due applicazioni può essere attiva simultaneamente. Al momento della stesura del documento l'applicazione in funzione risponde all'indirizzo:

**test-dapp.energychain.it** e verrà messa in produzione su indicazione specifica.

## Certificati SSL

I certificati SSL sono stati creati tramite il servizio gratuito certbot:

<https://certbot.eff.org/lets-encrypt/ubuntuionic-apache.html>

## SMTP

Per permettere la registrazione degli utenti tramite verifica dell'indirizzo email si è reso necessario attivare un servizio di email transazionale tramite protocollo SMTP. Si è scelto il servizio gratuito Sendinblue: <https://www.sendinblue.com/>

I parametri necessari nonché le credenziali per poter inviare le email sono presenti nei file di configurazione predisposti, si riportano i parametri principali di seguito

<b>Provider</b>	Sendinblue
-----------------	------------

<b>Indirizzo Host</b>	smtp-relay.sendinblue.com
<b>Posrta</b>	587
<b>Utente</b>	dapp@energychain.it
<b>TLS</b>	Abilitato
<b>Max email / giorno</b>	300
<b>Max email / ora</b>	100

# Appendice A - Esempio di PDF con Sigillo Lambda

## **SIN\_00968 - THE LEARNING METERS NETWORK** SMART CITIES AND COMMUNITIES AND SOCIAL INNOVATION

---



Public keys enabled list:

0xC7664a813055340E6e938CBebb62159661c37D03

0xF55cAfb2782A2E5F4170E0BC3708Af4cD420c13

0x0cc31107048C72DEd5384DB274AE7d664772Cd6b

0x113B05Fd11ef144ABC4eb399B97Ff5f55bdf7f0



*Ministero dell'Università e della Ricerca*

AMBITO DEL PROGETTO:  
MIUR - D.D. 5 luglio 2012 n. 391/Ric.  
SMART CITIES AND COMMUNITIES  
AND SOCIAL INNOVATION

WP formativo del SCN\_00398 – CUP J49G14000140008

CAPOFILA SIN: Università degli Studi di Perugia  
Dipartimento di INGEGNERIA

PROPONENTI SIN\_00968  
Danilo Guida, Alessandro Amirante,  
Gianluca Salamanno, Guido Ferrari Bravo



# Appendice B - Librerie, dipendenze e installazione

## Installazione Nodo Quadrans e dipendenze

```
$ wget http://repo.quadrans.io/installer/gqdc-installer.sh
$ sudo bash gqdc-installer.sh
```

Modificare /home/quadrans/gqdc.sh aggiungendo RPC se si tratta di un'installazione di Mainnet, altrimenti l'RPC è già configurato in testnet:

```
$ /usr/local/bin/gqdc --rpc --rpcapi eth,web3,net,personal,admin --rpccorsdomain "*"
--allow-insecure-unlock
```

```
$ sudo apt update
$ sudo apt install apache2 postgresql postgresql-contrib libapache2-mod-wsgi-py3 git
build-essential gettext python3-pip virtualenv python3-venv screen
```

## Clonare il repository:

```
$ mkdir -p /home/projects/energy-chain/
$ cd /home/projects/energy-chain/
$ git clone git@github.com:davidecosta/energy_meters.git
```

## Creazione, attivazione e installazione del virtual environment (tutte le librerie sono contenute nel file requirements.txt):

```
# cd /home/projects/energy-chain/
# python3 -m venv venv-energy-chain
# source /home/projects/energy-chain/venv-energy-chain/bin/activate
# cd /home/projects/energy-chain/energy_meters/
# sh dependencies.sh
# pip3 install wheel
# pip3 install -r requirements.txt
```

## Impostazioni finali

```
# mkdir /home/projects/energy-chain/energy_meters/media
# mkdir /var/log/apache2/energy/
# chown -R www-data: /var/log/apache2/energy/
# chown -R www-data: /home/projects/energy-chain/energy_meters/
# python manage.py migrate
# systemctl restart apache2
```

# Appendice C - Configurazione Server Web

## Esempio di configurazione

```
<IfModule mod_ssl.c>
<VirtualHost *:443>
    ServerName dapp.energychain.it

    ErrorLog ${APACHE_LOG_DIR}/energy/error.log
    CustomLog ${APACHE_LOG_DIR}/energy/access.log combined

    Alias /static/ /home/projects/energy-chain/energy_meters/static/
    <Directory /home/projects/energy-chain/energy_meters/static>
    Options FollowSymLinks MultiViews
    Require all granted
    </Directory>

    Alias /static/ /home/projects/energy-chain/energy_meters/static/
    <Directory /home/projects/energy-chain/energy_meters/static>
    Options FollowSymLinks MultiViews
    Require all granted
    </Directory>

    Alias /media/ /home/projects/energy-chain/energy_meters/media/
    <Directory /home/projects/energy-chain/energy_meters/media>
    Options FollowSymLinks MultiViews
    Require all granted
    </Directory>

    WSGIDaemonProcess energychain
    WSGIProcessGroup energychain
    WSGIApplicationGroup %{GLOBAL}
    WSGIScriptAlias / /home/projects/energy-chain/energy_meters/index.wsgi

    <Directory /home/projects/energy-chain/energy_meters/>
    Require all granted
    </Directory>

    SSLCertificateFile /etc/letsencrypt/live/dapp.energychain.it/fullchain.pem
    SSLCertificateKeyFile /etc/letsencrypt/live/dapp.energychain.it/privkey.pem
    Include /etc/letsencrypt/options-ssl-apache.conf
</VirtualHost>
</IfModule>
```

Autorizza, l'Università degli Studi di Perugia e i proponenti del progetto SIN\_00968, senza limiti di tempo, anche ai sensi degli artt. 10 e 320 cod.civ. e degli artt. 96 e 97 legge 22.4.1941, n. 633, Legge sul diritto d'autore, alla pubblicazione, modifica e/o diffusione in qualsiasi forma del presente elaborato e prende atto che la finalità di tali pubblicazioni sono meramente di carattere informativo ed eventualmente promozionale.