



Ricerca di Sistema elettrico

Qualificazione della infrastruttura e dei servizi per le LEC

G. D'Agosta, C. Meloni,
A. Annunziato, S. Branchetti, G. D'Agostino, P. De Sabbata,
N. Gessa, G. Massa, F. Moretti, G. Nigliaccio, C. Petrovich,
S. Romano, F. Paolucci, S. Pizzuti, A. Tofani, M. Caldera, M. Chinnici

Qualificazione della infrastruttura e dei servizi per le LEC

G. D'Agosta, C. Meloni, A. Annunziato, S. Branchetti, G. D'Agostino, P. De Sabbata, N. Gessa, G. Massa, F. Moretti, G. Nigliaccio, C. Petrovich, S. Romano, F. Paolucci, S. Pizzuti, A. Tofani, M. Caldera, M. Chinnici

Dicembre 2021

Report Ricerca di Sistema Elettrico

Accordo di Programma Ministero della Transizione Ecologica - ENEA

Piano Triennale di Realizzazione 2019-2021 - III annualità

Obiettivo: Tecnologie

Progetto: Tecnologie per la penetrazione efficiente del vettore elettrico negli usi finali

Work package: Local Energy District

Linea di attività: 1.49 Qualificazione della infrastruttura e dei servizi per le local energy communities

Responsabile del Progetto: Claudia Meloni, ENEA

Responsabile del Work package: Claudia Meloni, ENEA

Gli autori desiderano ringraziare il dott. Francesco De Candia laureando del Politecnico di Torino per il prezioso contributo allo sviluppo del simulatore basato sul modello ad agenti nel corso del suo tirocinio curricolare presso ENEA.

Indice

SOMMARIO	6
INTRODUZIONE	7
1 QUALIFICAZIONE SERVIZIO RECON	8
1.1 PROGETTAZIONE E IMPLEMENTAZIONE MOTORE DI CALCOLO	8
1.2 IMPLEMENTAZIONE DEL FRONTEND	9
1.3 HOMEPAGE	10
1.4 LE MIE SCHEDE	10
1.5 CREAZIONE/MODIFICA SCHEDA.....	12
1.6 REPORT DEI RISULTATI	15
1.7 IMPLEMENTAZIONE DEL BACKEND	16
1.8 INPUT.....	17
1.9 OUTPUT	20
2 GOVERNANCE DELLA LOCAL TOKEN ECONOMY	23
2.1 IL MODELLO AD AGENTI SMART TOKEN COMMUNITY	29
2.1.1 <i>Indagine preliminare sulle teorie comportamentali per il modello ad agenti</i>	29
2.1.2 <i>La teoria del comportamento previsto (theory of planned behavior)</i>	30
2.1.3 <i>La teoria dell'accordo relativo (relative agreement)</i>	31
2.1.4 <i>La struttura del modello ad agenti</i>	31
2.1.5 <i>I parametri del modello</i>	33
2.1.6 <i>Le classi del modello</i>	33
2.1.6.1 Scheduler.....	33
2.1.6.2 Community.....	34
2.1.6.3 Citizens	34
2.1.6.4 Services.....	36
2.1.6.5 Smart token economy	36
2.1.7 <i>Visualizzazione dei grafici e GUI MESA</i>	36
2.1.8 <i>Analisi di sensitività sui parametri del modello del Relative Agreement</i>	40
2.1.8.1 <i>Analisi di sensitività del parametro ϕ</i>	41
2.1.8.2 <i>Analisi di sensitività del parametro μ</i>	41
3 QUALIFICAZIONE MARKETPLACE	43
4 QUALIFICAZIONE ECLISTENER	46
4.1 QUALIFICAZIONE DEL CLUSTER	46
5 QUALIFICAZIONE DEI MODELLI PER L'ELABORAZIONE DEI DATI.....	55
5.1 CONSUMO GIORNALIERO.....	56
5.2 ANOMALIE – MONITORAGGIO DEI DISPOSITIVI NON ATTIVI	60
5.3 ANOMALIE/FUNZIONAMENTO – MONITORAGGIO DEI DISPOSITIVI FUNZIONANTI	61
5.4 DATI DI IRRAGGIAMENTO SOLARE.....	63
5.5 CONFRONTO TRA PRODUZIONE E CONSUMO DI ENERGIA	64
5.6 CONSUMO MENSILE	65
5.7 APPROCCIO ALLA VISUALIZZAZIONE DEI DATI.....	67
6 CONCLUSIONI	69
RIFERIMENTI BIBLIOGRAFICI	70

Figure

Figura 1: Workflow della web application RECON	8
Figura 2. Processo di calcolo di una scheda	9
Figura 3. Homepage di RECON	10
Figura 4. Sezione "Le mie schede".....	11
Figura 5. Esportazione lista schede	11
Figura 6. Filtraggio schede.....	12
Figura 7: Sezione 1 - Dati generali	13
Figura 8. Sezione 2 - Unità di consumo	13
Figura 9. Sezione 3 - Impianto di produzione.....	14
Figura 10. Sezione 4 - Parametri economici e finanziari	14
Figura 11. Sezione calcolo dei risultati	15
Figura 12. Report risultati.....	16
Figura 13. Diagramma ER del Database MySQL	17
Figura 14. Grafico con l'analisi energetica mensile calcolata da RECON.....	22
Figura 15. Grafico con i flussi di cassa attualizzati calcolati da RECON	22
Figura 16: Flusso dei dati in input e output tra simulatore e amministratore LTE.....	23
Figura 17: Estratto della GUI del modello ad agenti "Smart Token Economy"	33
Figura 18: Griglia degli agenti della GUI di MESA.....	37
Figura 19: Andamento degli agenti in base alla loro adesione alla comunità.....	37
Figura 20: Andamento medio dell'opinione e della relativa incertezza degli agenti esterni alla comunità.....	38
Figura 21: Confronto tra l'andamento cumulato dei token scambiati, delle transazioni e delle tasse nella comunità	39
Figura 22: Andamento delle transazioni e dei token scambiati ad ogni intervallo di tempo	39
Figura 23: Andamento delle transazioni per i vari servizi attivi tra cittadini e verso la comunità.....	39
Figura 24: Andamento a livello della comunità degli ingressi (total fee, euro fund, fondo di solidarietà) e delle uscite (total bonus adesione, servizi alla comunità), in termini sia di token che di euro.....	39
Figura 25: Sensitività del modello al parametro ϕ	41
Figura 26: Andamento del parametro ϕ per i cittadini esterni (non attivi) e interni (attivi) alla comunità	41
Figura 27: Sensitività del modello al parametro μ	42
Figura 28: Andamento del parametro μ per i cittadini esterni (non attivi) e interni (attivi) alla comunità	42
Figura 29: Andamento dei token di un utente	43
Figura 30: variazione dei token in base alla sottoscrizione dei contratti nel tempo.....	44
Figura 31: Dettaglio dei bonus ricevuti	44
Figura 32: Pubblicazione offerta e dashboard di gestione e vista dell'ENEA Marketplace	45
Figura 33: Login della dashboard di Kubernetes	48
Figura 34: Dashboard Kubernetes. Panoramica generale delle risorse consumate dal cluster	49
Figura 35: Dashboard Kubernetes. Risorse consumate dai pods dell'applicazione MediaBot.....	49
Figura 36: Esempio output del comando kubectl top abilitato da metric-server.....	50
Figura 37: Dashboard Prometheus.....	51
Figura 38: Architettura applicazione MediaBot	52
Figura 39: Architettura in produzione dell'applicazione TwitterBot	54
Figura 40: "Avatar" del cluster kubernetes sulla piattaforma cloud "open-stack" "Arena".	55
Figura 41: Esempio di consumo giornaliero	56
Figura 42: Esempio di profilo di carico medio	58
Figura 43: Esempio di profilo di carico istantaneo	59
Figura 44: Esempio di monitoraggio di dispositivi attivi.....	60
Figura 45: Esempio di monitoraggio del numero di misurazioni.....	62
Figura 46: Esempio di grafico per i dati di irraggiamento solare.....	63

Figura 47: Esempio di grafico di produzione da FV	64
Figura 48: Esempio di consumo mensile per uno specifico utente	66

Tabelle

Tabella 1. Sezioni più rilevanti del frontend	9
Tabella 2: parametri di controllo per l'amministratore della LTE.....	24
Tabella 3: Indicatori economici	28
Tabella 4: Indicatori energetico-ambientali	28
Tabella 5: Indicatori sociali	28
Tabella 6: Parametri iniziali del modello per l'analisi di sensitività	40
Tabella 7: definizione grandezze	56

Sommario

Il documento presenta i risultati della fase di qualificazione e valutazione dei differenti prodotti software e modelli sviluppati nella attività 1.49 del progetto.

L'obiettivo di questa attività è comprendere il livello di qualità dei servizi realizzati nel periodo precedente e quali potrebbero essere le prospettive di miglioramento.

Il documento presenta inizialmente la valutazione dello strumento RECON, che permette di identificare il valore economico dell'utilizzo di sorgenti da fonte rinnovabile all'interno di un progetto specifico di comunità energetica, valutando sia il ritorno economico che altri parametri economici a partire da poche informazioni legate al consumo elettrico e alla configurazione degli impianti. Il tool, per il momento, utilizza i pannelli fotovoltaici come sorgente ma è previsto, in un prossimo futuro, l'estensione anche ad altre tipologie di sorgenti.

Successivamente, viene presentata l'analisi fatta per i modelli di governance della token economy. Poiché gli strumenti informatici necessari per questo aspetto non sono ancora disponibili, vengono presentati dei modelli matematici che permettono di capire il funzionamento dinamico di questo tipo di economia.

Nel capitolo 3 viene presentato un modello di market place basato su un prodotto commerciale per dimostrare la fattibilità e l'usabilità di uno strumento di questo tipo all'interno delle dinamiche della Local Energy Community. La scelta di utilizzare uno strumento commerciale è dovuta all'alto costo di sviluppo che questo tipo di strumenti richiedono per raggiungere un livello di sicurezza sufficiente.

Nel capitolo 4 viene presentato il lavoro di qualificazione del progetto ECListener per il monitoraggio del *sentiment* della Comunità rispetto agli argomenti di trend e alle parole chiave utilizzati sui social network da parte dei membri della comunità stessa. Tale strumento permetterà al gestore di capire quali sono gli interessi della comunità e quali opinioni sono presenti rispetto determinati argomenti.

Nel capitolo 5 vengono presentati alcuni modelli matematici per l'analisi dei dati e le relative interrogazioni al database dei consumi e della produzione che forniscono le informazioni richieste. Questi modelli sono alla base del sistema di gestione proposto per l'amministratore della Comunità che, attraverso un set di Key Performance Indicators e gli strumenti di confronto offerti dal cruscotto per il gestore della comunità, chiamato CruISE.

Le conclusioni, al capitolo 6 chiudono il presente report, indicando i futuri sviluppi degli strumenti presentati.

Introduzione

Lo sviluppo dei tool a supporto delle Local Energy Community procede con la fase di validazione dei risultati e la qualificazione in contesti controllati.

L'obiettivo di questa fase è duplice: da una parte integrare le componenti per fornire un servizio unitario e completo, nell'ottica della gestione del ciclo di vita della comunità presentato nel precedente report, dall'altra quella di identificare le potenzialità degli strumenti sviluppati nei differenti contesti.

Seguendo il concetto di *ciclo di vita della Comunità Energetica*, ENEA ha sviluppato una serie di strumenti che coprono differenti aspetti di questo ciclo di vita, dalla progettazione e studio di fattibilità fino alla gestione e alla valutazione del livello di gradimento o di interesse, passando per il monitoraggio delle abitazioni e della produzione.

Questi strumenti sono stati progettati e successivamente ingegnerizzati e sviluppati per rispondere alle esigenze delle nasciture Local Energy Community sul territorio nazionale attraverso un'analisi delle possibili necessità partendo dall'esperienza accumulata da ENEA e dagli altri partner nelle iniziative sviluppate negli ultimi anni legate alla gestione del territorio e delle Smart Communities.

Il presente documento presenta i risultati della qualificazione degli strumenti prodotti nelle fasi precedenti del progetto e che dovranno subire una fase di test e valutazione prima di essere resi disponibili alle persone interessate alla realizzazione di una Local Energy Community.

Il documento si focalizza sulla qualificazione dei seguenti tool:

- Lo strumento RECON per l'analisi di fattibilità economica della comunità;
- I modelli di gestione e valorizzazione della token economy;
- L'utilizzo di uno strumento commerciale per la realizzazione di un market place delle iniziative;
- Lo strumento ECListener di analisi del *sentiment* della comunità;
- I modelli dati e le funzioni di estrazione dei dati da utilizzare nel cruscotto CruISE per il gestore della comunità.

L'obiettivo di questa fase di qualificazione è capire e monitorare la qualità dei servizi sviluppati durante il periodo di svolgimento della linea di attività 1.49 e immaginare le potenzialità di questi strumenti all'interno di un ecosistema complesso come quello della Local Energy Community.

Nell'ultimo periodo, l'evoluzione della normativa e delle tecnologie collegate alle comunità dell'energia ha portato alla fioritura di iniziative commerciali e sperimentali che hanno l'obiettivo di realizzare delle comunità energetiche e, dall'altra parte, di fornire servizi commerciali agli stakeholder, tra cui comuni e privati cittadini, che vedano l'investimento in fonti rinnovabili come una possibile soluzione alle proprie esigenze di energia. Gli strumenti proposti da ENEA hanno l'obiettivo di rappresentare un riferimento per il mercato, mostrando quali possono essere le possibilità e gli sviluppi di queste tecnologie sganciandosi dalla mera necessità commerciale ma proponendo, per alcuni strumenti, anche delle idee per il futuro. Per questo motivo, laddove il mercato proporrà soluzioni tecnologiche simili o comparabili con quelle proposte da ENEA, queste ultime verrebbero comunque non ulteriormente sviluppate, con l'obiettivo specifico di non creare problematiche al mercato.

Un ulteriore lavoro di integrazione dei differenti servizi proposti deve essere sviluppato nel prossimo futuro, al fine di avere un approccio coordinato ed identificabile agli stakeholder e ai cittadini, e garantire una qualità sufficiente ai servizi proposti.

1 Qualificazione servizio RECON

Il presente paragrafo illustra il risultato dello sviluppo dello strumento RECON e descrive il funzionamento del motore di calcolo.

1.1 Progettazione e implementazione motore di calcolo

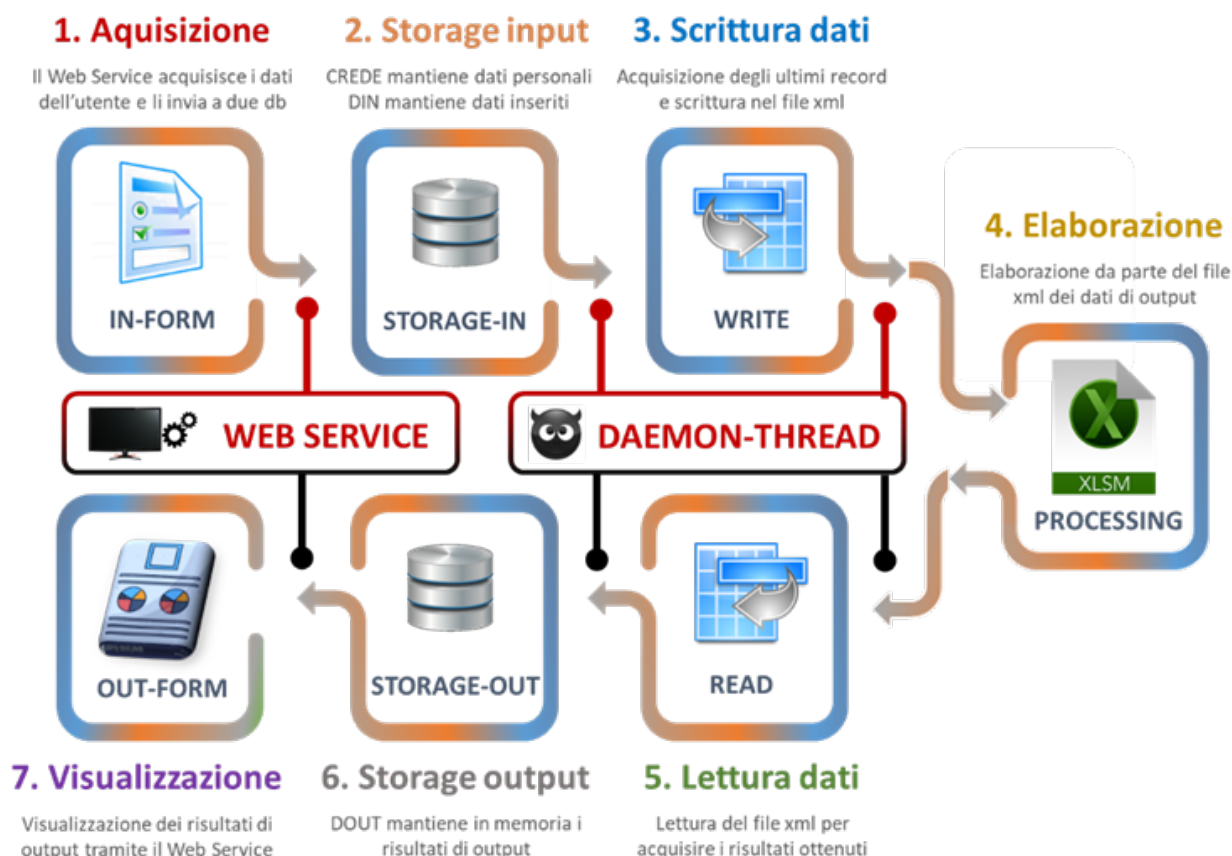


Figura 1: Workflow della web application RECON

Il motore di calcolo è basato su un simulatore Excel, e prevede una serie di input necessari che saranno richiesti all'utente tramite la GUI. Tali input sono processati dall'Excel attraverso opportune formule e macro Visual Basic. Al fine di automatizzare il processo di inserimento e lettura dati da e verso Excel, è stato sviluppato un wrapper che si interfaccia con Excel.

Come mostrato in Figura 1, il motore di calcolo è disaccoppiato dalla web application, sia in termini logici che fisici, in quanto è ospitato su una differente macchina virtuale ed è costituito dal simulatore Excel e da un wrapper.

Il wrapper è stato sviluppato in Python, in quanto le librerie di interfacciamento con Excel messe a disposizione dal repository pubblico hanno consentito la manipolazione senza particolari problemi, a differenza delle librerie precedentemente testate in altri linguaggi (es. Java e NodeJS).

Le librerie PyWin32 sono disponibili nel repository pubblico di Python, PiP, e consentono di gestire programmaticamente i file Excel: apertura, chiusura, inserimento e lettura dati dalle celle ed esecuzione di macro. La fruizione del calcolo tramite il wrapper da parte della web application è garantita da un servizio RES messo a disposizione dal wrapper tramite il web server Flask.

La Figura 2 mostra il dettaglio della comunicazione tra le entità durante il processo di richiesta di calcolo di una scheda da parte dell'utente:

1. l'utente richiede il calcolo di una scheda;
2. Laravel gestisce la richiesta recuperando i metadati della scheda dal DB MySQL, in particolare lo sheet_uuid, identificativo univoco della scheda in MongoDB;
3. Il controller di Laravel invia una chiamata HTTP Post al servizio REST del computation engine Python, passando come parametro lo sheet_uuid;
4. il computation engine recupera da MongoDB la i dati della scheda;
5. il computation engine inserisce i dati di input del simulatore in maniera automatica all'interno di Excel ed avvia le macro per il calcolo;
6. il computation engine recupera i dati di output dall'Excel e li scrive su MongoDB;
7. la chiamata REST restituisce un valore di ritorno positivo nel caso in cui il calcolo sia andato a buon fine, altrimenti negativo;
8. i dati sono disponibili per essere mostrati a frontend tramite Laravel.

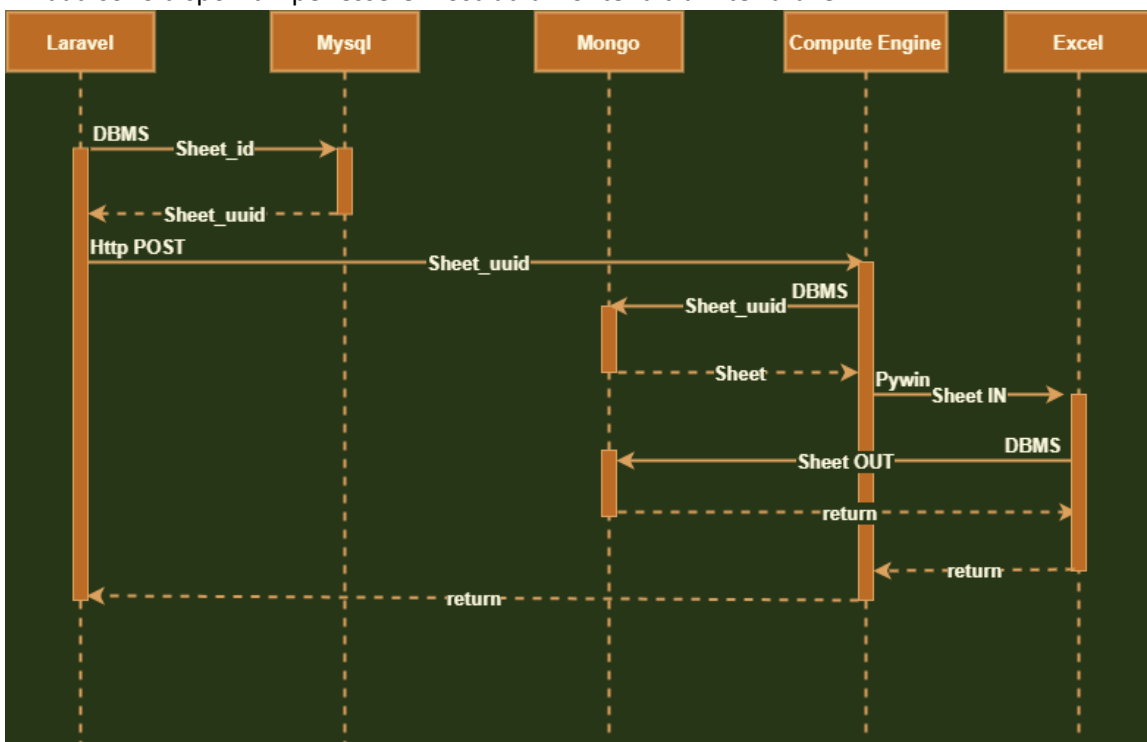


Figura 2. Processo di calcolo di una scheda

1.2 Implementazione del frontend

L'implementazione del frontend è stata eseguita sulla base della progettazione precedentemente descritta e possiamo suddividerla in diverse sezioni dell'applicazione web. Ogni sezione ha una tipologia di accesso dipendente dalla modalità utilizzata dall'utente: accesso libero, registrato o amministratore.

Tabella 1. Sezioni più rilevanti del frontend

Pagina	Accesso
Homepage	Libero
Il simulatore	Libero
Il logo	Libero
Le mie schede	Utente registrato
Nuova scheda	Utente registrato
Tabella utenti	Amministratore
Contatti	Libero
Registrazione	Libero

Report risultati	Utente registrato
------------------	-------------------

1.3 Homepage

Nella homepage è possibile fruire del video informativo e della descrizione dello strumento: cosa è possibile ottenere, quali informazioni sono richieste ed una procedura step-by-step su cosa è necessario fare per finalizzare la simulazione (Figura 3).

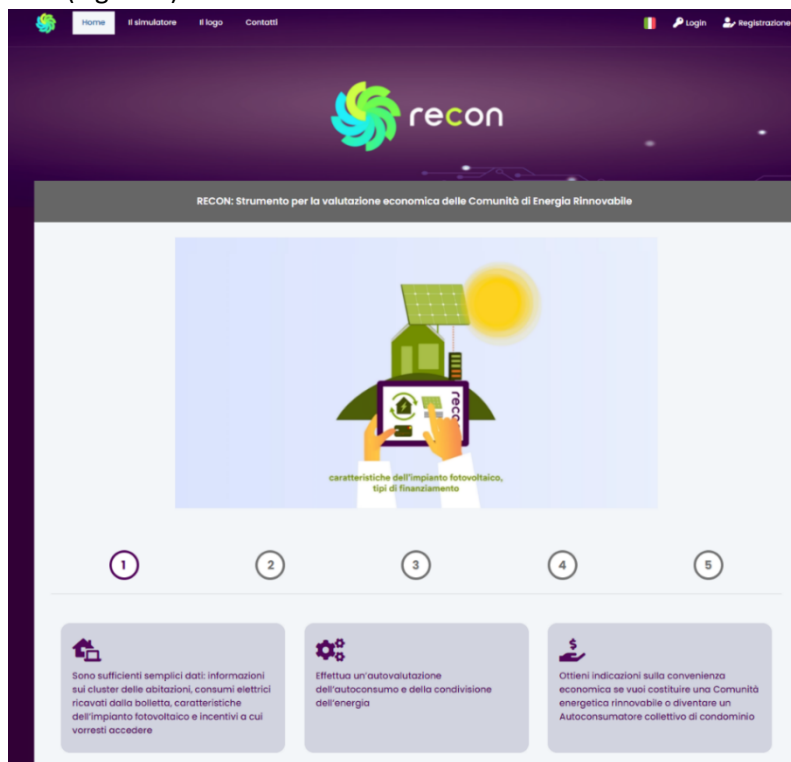






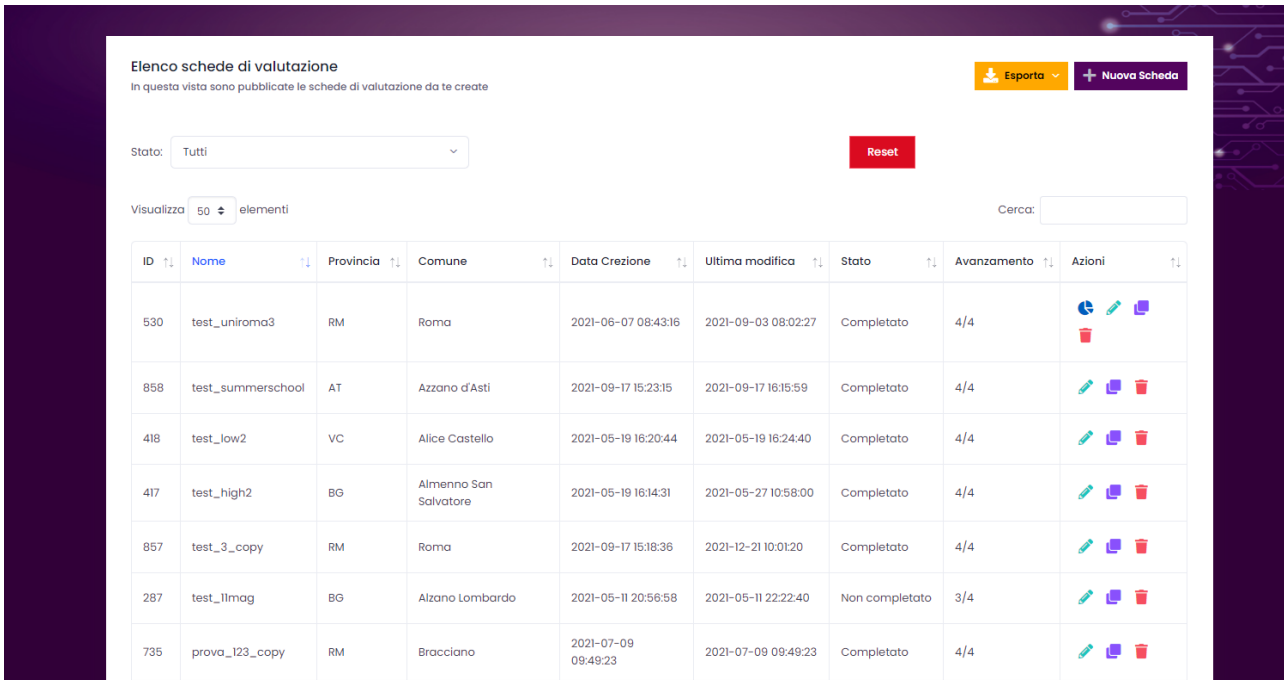
Figura 3. Homepage di RECON

1.4 Le mie schede

Nella sezione “Le mie schede” è possibile gestire le schede inserite dall’utente e crearne di nuove (Figura 4). La pagina mostra tutte le schede comprensive di Id, nome scheda, data creazione, data ultima modifica, stato (completato/non completato).

- ID. Identificativo univoco della scheda autogenerato dal sistema.
- Nome. Nominativo della scheda definito dall’utente.
- Provincia. Provincia di riferimento della scheda.
- Comune. Comune di riferimento della scheda.
- Data creazione/modifica.
- Stato. Stato della scheda (completato/non completato)
- Avanzamento. Indica lo stato di avanzamento della compilazione della scheda: il numero indicato indica l’ultimo step completamente compilato e salvato.
- Azioni. Azioni possibili sulla scheda:
 1.  Eliminazione di una scheda.
 2.  Clonazione di una scheda.

3.  Editing scheda.
4.  Visualizzazione report. Questa funzione è abilitata solo se la scheda è nello stato “Completato”, e il calcolo della simulazione su tale scheda è stato correttamente eseguito.



Elenco schede di valutazione
In questa vista sono pubblicate le schede di valutazione da te create

Stato: Reset

Visualizza elementi Cerca:
















ID	Nome	Provincia	Comune	Data Creazione	Ultima modifica	Stato	Avanzamento	Azioni
530	test_uniroma3	RM	Roma	2021-06-07 08:43:16	2021-09-03 08:02:27	Completato	4/4	  
858	test_summerschool	AT	Azzano d'Asti	2021-09-17 15:23:15	2021-09-17 16:15:59	Completato	4/4	 
418	test_low2	VC	Alice Castello	2021-05-19 16:20:44	2021-05-19 16:24:40	Completato	4/4	 
417	test_high2	BG	Almenno San Salvatore	2021-05-19 16:14:31	2021-05-27 10:58:00	Completato	4/4	 
857	test_3_copy	RM	Roma	2021-09-17 15:18:36	2021-12-21 10:01:20	Completato	4/4	 
287	test_1lmag	BG	Alzano Lombardo	2021-05-11 20:56:58	2021-05-11 22:22:40	Non completato	3/4	 
735	prova_123_copy	RM	Bracciano	2021-07-09 09:49:23	2021-07-09 09:49:23	Completato	4/4	 

Figura 4. Sezione "Le mie schede"

È inoltre possibile esportare la lista in formato CSV o PDF tramite l'apposita funzione di esportazione, Figura 5.

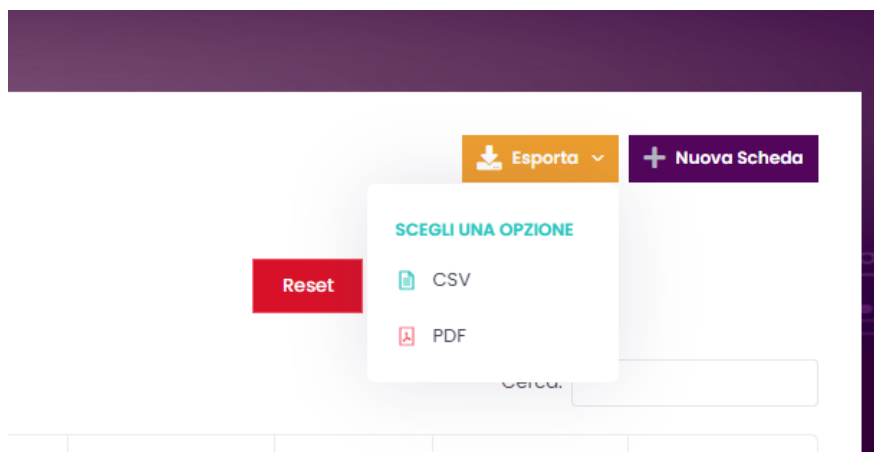


Figura 5. Esportazione lista schede

Nella casella di ricerca è possibile filtrare una scheda specifica per testo. Tale ricerca è di tipo full-text su tutte le stringhe del record, Figura 6.

The screenshot shows a web interface for managing records. At the top, there is a filter for 'Stato' set to 'Tutti' and a 'Reset' button. Below that, a 'Visualizza' dropdown is set to '50 elementi' and a search bar contains the text 'prova'. The main table has the following data:

ID	Nome	Provincia	Comune	Data Creazione	Ultima modifica	Stato	Avanzamento	Azioni
735	prova_123_copy	RM	Bracciano	2021-07-09 09:49:23	2021-07-09 09:49:23	Completato	4/4	[Pencil icon] [Trash icon]
734	prova_123	RM	Bracciano	2021-07-09 09:37:39	2021-07-09 09:48:50	Completato	4/4	[Pencil icon] [Trash icon]

Figura 6. Filtraggio schede

1.5 Creazione/modifica scheda

La sezione della creazione o modifica di una scheda prevede l'immissione dei dati suddivisi in quattro sezioni distinte, come descritto precedentemente nella sezione dei dati richiesti.

1. Dati generali (Figura 7)
2. Unità di consumo (Figura 8)
3. Impianto di produzione (Figura 9)
4. Parametri economici e finanziari (Figura 10)

Ogni campo è rappresentato in diverse modalità:

- testo libero;
- numerico;
- lista predefinita;
- scelta multipla;
- booleano (on-off).

Per facilitare l'inserimento dei dati minimizzando le possibilità di errore è stato introdotto un controllo interattivo dei campi, oscurando le scelte precluse dalle impostazioni date dall'utente. I campi impostati correttamente sono etichettati con colore verde ed una spunta, mentre quelli errati con un colore rosso ed una x.

Scheda

1. Dati generali → 2. Unità di consumo → 3. Impianto di produzione → 4. Parametri economici e finanziari → 5. Risultati

Sezione 1

Informazioni scheda

Nome Scheda

Solo all'animerici spazio e sottotetto

Località

Provincia Comune

Configurazione selezionata

Autoconsumatore collettivo
 Comunità di energia rinnovabile
 Nessuna

Tipologia di connessione alla rete dell'impianto di produzione FER
 Bassa tensione Media tensione

SUCCESSIVO

Figura 7: Sezione 1 - Dati generali

Scheda **Test**

1. Dati generali → 2. Unità di consumo → 3. Impianto di produzione → 4. Parametri economici e finanziari → 5. Risultati

Sezione 2

Utenze comuni collegate all'impianto di produzione da fonte rinnovabile

Presenza utenze comuni collegate all'impianto di produzione da fonte rinnovabile

no

Utenze domestiche

Cluster di abitazioni

N° cluster

Cluster N° 1

Nome cluster [1] N° abitazioni nel cluster [1]

Dati sull'occupazione [1]

Numero di occupanti nell'abitazione-tipo [1] Numero di occupanti nelle ore diurne nell'abitazione-tipo [1]

Figura 8. Sezione 2 - Unità di consumo

Scheda **Test**

1. Dati generali → 2. Unità di consumo → 3. Impianto di produzione → 4. Parametri economici e finanziari → 5. Risultati

Sezione 3

Impianto fotovoltaico

Potenza installata kW

Tecnologia delle celle fotovoltaiche

Tipologia di installazione dei moduli FV

Inclinazione dei moduli FV Gradi

Orientamento dei moduli FV Gradi

Perdita % media annuale di rendimento %

PRECEDENTE SUCCESSIVO

Figura 9. Sezione 3 - Impianto di produzione

Scheda **Test**

1. Dati generali → 2. Unità di consumo → 3. Impianto di produzione → 4. Parametri economici e finanziari → 5. Risultati

Sezione 4

Parametri economici €

Investimento unitario iniziale Euro/kWp

Costi O&M annuali Euro/anno

Prezzo medio di acquisto dell'energia elettrica Euro/kWh

Prezzo medio di vendita dell'energia prodotta dall'impianto FV e immessa in rete Euro/kWh

Utilizzo delle detrazioni fiscali

Bonus Risparmio energetico (detrazione 50%)

Da considerare no

Superbonus (detrazione 110%)

Da considerare no

Parametri finanziari

Tasso di sconto Inflazione Periodicità della manutenzione straordinaria

Figura 10. Sezione 4 - Parametri economici e finanziari

La sezione finale dei risultati (Figura 11), consente il calcolo della scheda e l'abilitazione della pagina degli output, previo esito positivo della simulazione. È possibile effettuare anche altre azioni ausiliarie quali l'esportazione dei dati di input in formato JSON o tornare alla lista delle schede inserite.

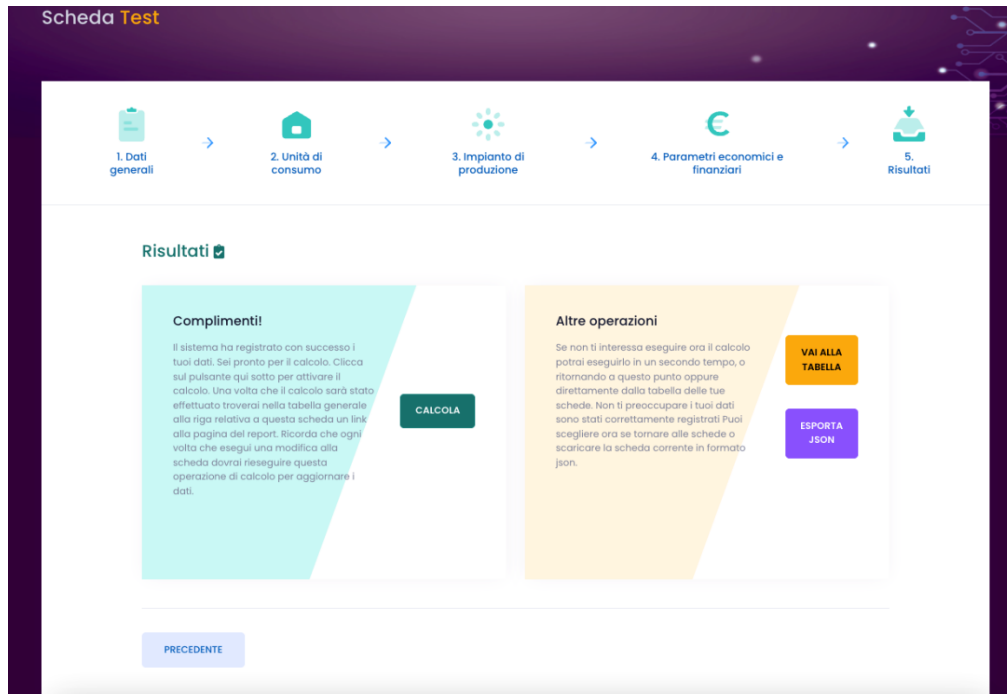


Figura 11. Sezione calcolo dei risultati

1.6 Report dei risultati

Il report dei risultati (Figura 12) mostra gli output descritti nel paragrafo (Inserire paragrafo output) e consente:

- la visualizzazione online dei dati;
- l'esportazione in formato JSON;
- l'esportazione in formato PDF.

L'esportazione in formato JSON è poco user friendly ma eventualmente utilizzabile come input per elaborazioni dati di terze parti, mentre l'esportazione in PDF, descritta nel paragrafo **Errore. L'origine riferimento non è stata trovata.**, consente di avere un documento strutturato consultabile anche offline.

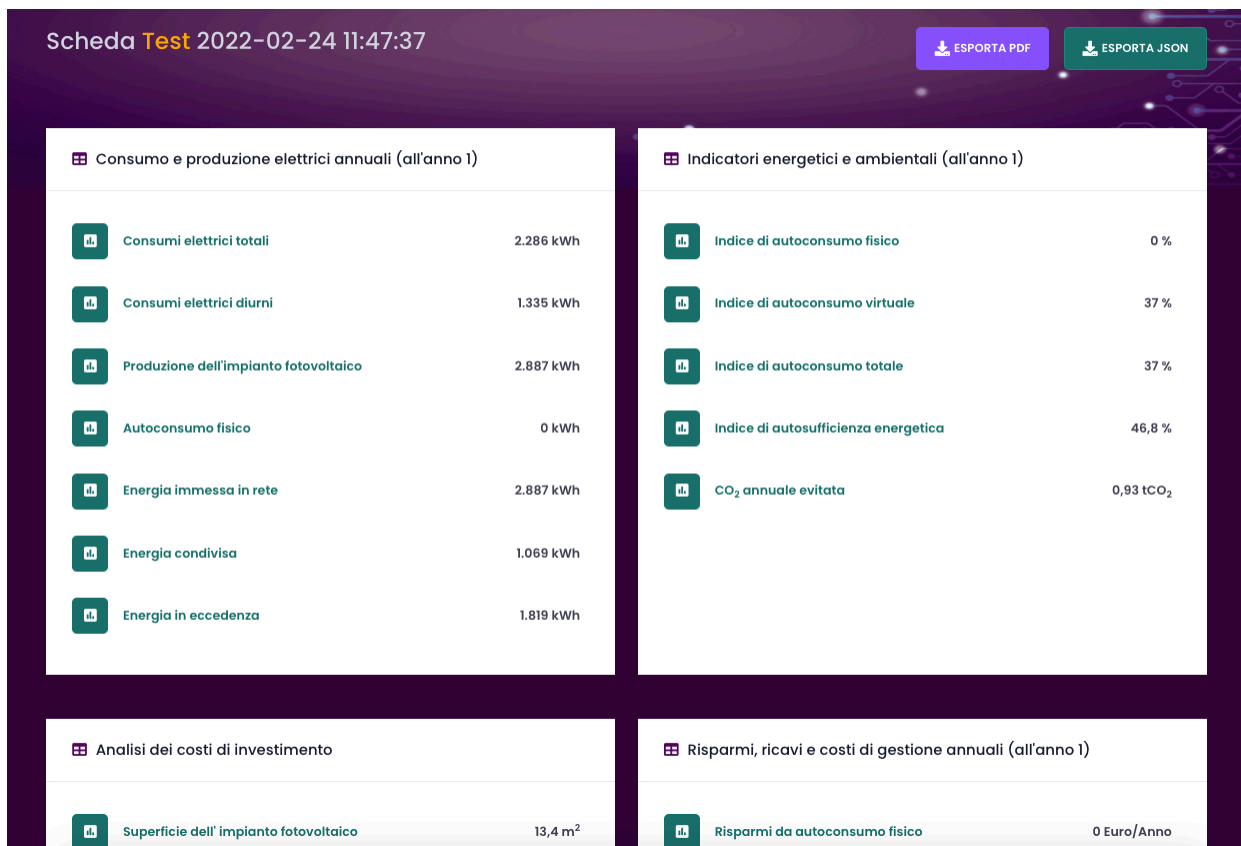


Figura 12. Report risultati

1.7 Implementazione del backend

Il backend è stato implementato utilizzando due sistemi di persistenza complementari: MySQL e MongoDB. Il DB MySQL è utilizzato per l’immagazzinamento delle tabelle ausiliari di Laravel, per le informazioni sull’utente e per i metadati delle schede.

La Figura 13 mostra il diagramma ER (Entity-Relationship) del Database MySQL utilizzato, attraverso cui è possibile visualizzare il dettaglio del contenuto informativo delle tabelle.

Le tabelle ausiliari di Laravel sono:

- migrations;
- model_has_permissions;
- model_has_roles;
- failed_jobs;
- permissions;
- roles;
- role_has_permissions;
- personal_access_tokens.

Le tabelle **users** e **sheets** sono invece tabelle create per la persistenza delle informazioni sugli utenti e delle informazioni riguardanti i metadati delle schede. I dati delle schede sono conservati nel database MongoDB. Il database MongoDB contiene le informazioni delle schede in due “collection” distinte: **sheets_in**, contenente i dati di input, e **sheets_out**, contenente i dati di output.

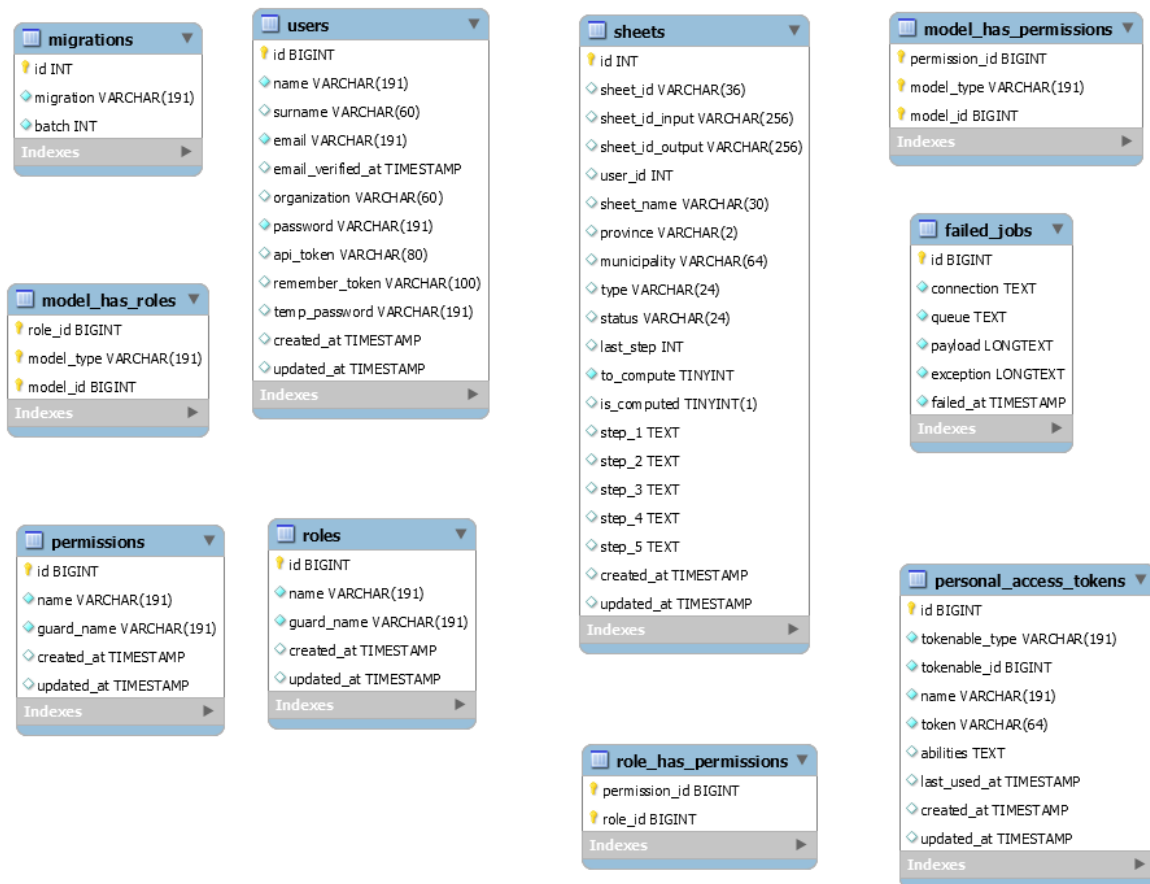


Figura 13. Diagramma ER del Database MySQL

1.8 Input

Di seguito sono riportati i dati di input che si presentano all'utente tramite l'interfaccia di RECON.

- Nome della scheda di valutazione
- Configurazione
 - o Tipologia della configurazione: comunità energetica, autoconsumatore collettivo oppure nessuna delle due precedenti (opzione che può essere utilizzata per valutare il contributo degli incentivi sull'energia condivisa sui risultati economico-finanziari)
 - o Tipologia di connessione alla rete pubblica dell'impianto fotovoltaico (solo per gli autoconsumatori collettivi): bassa tensione o media tensione
- Località: provincia e comune
- Unità di consumo:
 - o Utenze comuni collegate all'impianto di produzione alimentato a fonte rinnovabile: sono le utenze collegate allo stesso POD dell'impianto fotovoltaico, che quindi contribuiscono all'autoconsumo fisico
 - Presenza delle utenze comuni
 - Disponibilità dei consumi da bolletta (se le utenze comuni sono presenti): mensile, annuale, non disponibili
 - Disponibilità dei consumi in fascia F1 (SI/NO)
 - Consumi nel periodo [kWh] (anno o mese, se disponibili)
 - Consumi in fascia F1 [kWh] (nello stesso periodo temporale del dato precedente, se disponibili)

- Utenze domestiche non collegate all'impianto fotovoltaico, che rilevano per la condivisione dell'energia:

Numero di cluster di utenze domestiche. Per ciascun cluster sono richiesti i seguenti dati:

- Nome del cluster: utile per distinguere tra i vari cluster
- Numero di abitazioni nel cluster
- Occupanti nell'abitazione rappresentativa del cluster:
 - Numero di occupanti: utilizzato per la stima dei consumi elettrici per forza elettromotrice e per la produzione di ACS nel caso in cui non siano forniti i consumi elettrici mensili
 - Numero medio di occupanti presenti nelle ore diurne: utilizzato per la stima dei consumi elettrici nelle ore diurne nel caso in cui non siano forniti i consumi mensili in fascia F1
- Caratteristiche dell'abitazione rappresentativa del cluster:
 - Superficie riscaldata [mq]
 - Superficie raffrescata [mq]
 - Qualità termica dell'involucro: è una stima che tiene conto delle caratteristiche termiche dell'involucro (pareti verticali e orizzontali, serramenti). In presenza di un attestato di prestazione energetica è possibile considerare le seguenti corrispondenze:
 - qualità ottima → classi A1 e superiori
 - qualità buona → classe B
 - qualità sufficiente → classi C, D
 - qualità scarsa → classi E, F
 - qualità mediocre → classe G
- Caratteristiche degli impianti dell'abitazione rappresentativa del cluster:
 - Tipologia dei condizionatori: solo freddo o caldo-freddo (pompa di calore)
 - Combustibile dell'impianto di riscaldamento principale
 - Tipologia del generatore per l'ACS (nel caso di alimentazione elettrica): boiler elettrico o pompa di calore
- Tipologia di utenze elettriche dell'abitazione rappresentativa del cluster (presenza o meno):
 - Forza elettromotrice e illuminazione (sempre presente)
 - Raffrescamento (condizionamento estivo)
 - Riscaldamento
 - Produzione di ACS
- Quota percentuale dello specifico uso finale (in termini di fabbisogno utile di energia) coperto dall'energia elettrica (nel caso di alimentazione tramite energia elettrica dello specifico uso finale):
 - Raffrescamento (condizionamento estivo)
 - Riscaldamento
 - Produzione di ACS
- Consumi elettrici dell'abitazione rappresentativa del cluster
 - Contratto di energia elettrica monorario (SI/NO): nell'attuale versione questo dato non è valorizzato
 - Disponibilità dei consumi da bolletta: mensile, annuale, non disponibili
 - Disponibilità dei consumi in fascia F1 (SI/NO)

- Consumi nel periodo [kWh] (anno o mese, se disponibili)
- Consumi in fascia F1 [kWh] (nello stesso periodo del dato precedente, se disponibili)
- Impianto di produzione alimentato da fonte rinnovabile. Nell'attuale release di RECON è assimilato a un impianto fotovoltaico.
 - o Potenza di picco installata [kWp]
 - o Tecnologia delle celle fotovoltaiche: silicio cristallano, film sottile – CIS, film sottile – CdTe, sconosciuto. Nel caso sia indicato “sconosciuto”, RECON assume che le celle fotovoltaiche siano in silicio cristallino in quanto è la tecnologia attualmente più diffusa
 - o Tipologia di installazione dei moduli FV: installati sull'involucro dell'edificio (su edificio) o su apposita struttura metallica (free standing)
 - o Inclinazione dei moduli fotovoltaici rispetto al piano orizzontali [gradi] [0 +90]
 - o Orientamento dei moduli fotovoltaici [gradi]: angoli verso ovest sono positivi mentre quelli verso est sono negativi (0° sud, +90° ovest, -90° est)
 - o Perdita percentuale media annuale di rendimento [%]: è il fattore di derating (indicato nei datasheet dei moduli fotovoltaici) ed indica la perdita media annuale di efficienza dei moduli dovuta all'usura
- Parametri economici dell'investimento
 - o Investimento unitario iniziale [Euro/kW_p]: è il costo chiavi in mano di fornitura e installazione dell'impianto fotovoltaico, stimato o desumibile da un preventivo. Nel caso non sia noto è possibile lasciare il campo vuoto, RECON lo stima in base alla taglia dell'impianto utilizzando una curva desunta da indagini di mercato;
 - o Costi O&M annuali [euro/anno]: indica i costi annuali di manutenzione ordinaria e di gestione dell'impianto fotovoltaico, compresa la polizza assicurativa, e di gestione funzionamento della configurazione (es. eventuale consulenza di professionisti per l'espletamento delle pratiche e gli aspetti tecnici). È un dato obbligatorio che incide sensibilmente sull'analisi economico-finanziaria, pertanto va stimato con attenzione.
 - o Prezzo medio di acquisto dell'energia elettrica [euro/kWh]: indica il prezzo medio di acquisto dell'energia elettrica (IVA inclusa se applicabile) dell'utenza collegata allo stesso POD (di interfacciamento con la rete pubblica) dell'impianto fotovoltaico. Serve per il calcolo dei risparmi da autoconsumo fisico ed è un dato che può essere ricavato dalle bollette.
 - o Prezzo medio di vendita dell'energia prodotta dall'impianto fotovoltaico e immessa in rete [euro/kWh]: è pari al prezzo del ritiro dedicato GSE oppure, in caso di accordi di compravendita con un operatore di energia, al prezzo di vendita concordato contrattualmente.
 - o Detrazioni fiscali – bonus 50% risparmio energetico
 - Da considerare: SI/NO, nel caso in cui si opti per la detrazione 50% per l'acquisto dell'impianto fotovoltaico
 - Cessione del credito: SI/NO, nel caso in cui si opti per la cessione del credito in abbinamento con la detrazione 50%
 - Aliquota cessione del credito [%]: aliquota della cessione del credito, nel caso in cui si opti per la cessione del credito
 - Sconto in fattura: SI/NO, nel caso in cui si opti per lo sconto in fattura in abbinamento con la detrazione 50%. Lo sconto in fattura è alternativo alla cessione del credito

- Detrazioni fiscali – superbonus 110%
 - Da considerare: SI/NO, nel caso in cui si opti per il superbonus per l’acquisto dell’impianto fotovoltaico
 - Cessione del credito: SI/NO, nel caso in cui si opti per la cessione del credito in abbinamento con il superbonus
 - Aliquota cessione del credito [%]: aliquota della cessione del credito, nel caso in cui si opti per la cessione del credito
 - Sconto in fattura: SI/NO, nel caso in cui si opti per lo sconto in fattura in abbinamento con il superbonus. Lo sconto in fattura è alternativo alla cessione del credito
- Parametri finanziari
 - Tasso di sconto [%]: è il tasso di interesse utilizzato per attualizzare i flussi di cassa. Può essere assunto pari al rendimento offerto da attività finanziarie alternative all’investimento, caratterizzate da basso tasso di rischio e da lunga scadenza.
 - Inflazione [%]: tasso medio di inflazione da utilizzare per l’analisi finanziaria
 - Periodicità della manutenzione straordinaria [anni]: indica ogni quanti anni è previsto un intervento di manutenzione straordinaria dell’impianto fotovoltaico, ad esempio per la sostituzione dell’inverter
 - Finanziamento [euro]: ammontare dell’eventuale finanziamento (prestito bancario) attivato per coprire (anche parzialmente) l’investimento. Se non è previsto un finanziamento va posto pari a zero. Nel caso in cui, considerando anche il contributo delle detrazioni fiscali, risultasse che i costi di investimento residui siano inferiori al finanziamento indicato, il simulatore assume che il finanziamento sia pari ai costi di investimento residui
 - Tasso di interesse annuo [%]: tasso di interesse annuo del finanziamento. RECON assume che la restituzione del finanziamento avvenga tramite pagamenti costanti a un tasso di interesse costante (cd. ammortamento alla francese)
 - Durata del finanziamento [anni]: in presenza di finanziamento
 - Numero di rate annuali: in presenza di finanziamento

1.9 Output

Di seguito sono riportati gli output calcolati da RECON.

- Consumi e produzione elettrici annuali della comunità [kWh]
 - Consumi elettrici totali
 - Consumi elettrici diurni
 - Produzione dell’impianto fotovoltaico
 - Autoconsumo fisico
 - Energia condivisa
 - Energia immessa in rete
 - Energia in eccedenza
- Indicatori energetici e ambientali della comunità
 - Indice di autoconsumo fisico [%]: è il rapporto tra l’energia autoconsumata dalle utenze elettriche collegate allo stesso POD con la rete pubblica dell’impianto fotovoltaico e l’energia prodotta da quest’ultimo;
 - Indice di autoconsumo virtuale [%]: è il rapporto tra l’energia condivisa nella comunità e l’energia prodotta dall’impianto fotovoltaico;
 - Indice di autoconsumo totale [%]: è la somma degli indici di autoconsumo fisico e virtuale;

- Indice di autosufficienza energetica [%]: è il rapporto tra la somma dell'energia fisicamente autoconsumata e condivisa e il fabbisogno di energia elettrica della comunità;
- CO₂ annuale evitata [t_{CO2}]: è la quantità di CO₂ evitata dalla comunità per effetto dell'energia prodotta dall'impianto fotovoltaico (energia rinnovabile) e delle perdite di rete evitate grazie all'autoconsumo fisico e alla condivisione dell'energia nella comunità.
- **Analisi dei costi di investimento**
 - Superficie totale dei moduli fotovoltaici [m²]: superficie occupata dai moduli fotovoltaici, calcolata considerando la tipologia di celle fotovoltaiche;
 - Costo iniziale dell'impianto [euro]: costo chiavi in mano dell'impianto fotovoltaico, al netto di eventuali contributi in conto capitale a fondo perduto;
 - Investimento iniziale autofinanziato [euro]: costo dell'impianto che deve essere direttamente sostenuto dalla comunità per avviare il progetto (anno 0), calcolato come costo iniziale al netto dei contributi in conto capitale a fondo perduto, del finanziamento e delle detrazioni fiscali qualora per queste ultime si sia optato per la cessione del credito o per lo sconto in fattura.
 - Finanziamento [euro]: ammontare del finanziamento (qualora presente)
 - Superbonus [euro]: ammontare delle detrazioni beneficate con l'accesso al superbonus, indipendentemente dall'opzione scelta (detrazione diretta, sconto in fattura o cessione del credito);
 - Detrazione diretta annuale con superbonus [euro]: ammontare della detrazione diretta annuale relativa al superbonus (se utilizzato), qualora si sia optato per la detrazione diretta;
 - Bonus 50% risparmi energetici [euro]: ammontare delle detrazioni beneficate con l'accesso al bonus 50%, indipendentemente dall'opzione scelta (detrazione diretta, sconto in fattura o cessione del credito);
 - Detrazione diretta annuale con bonus 50% [euro]: ammontare della detrazione diretta annuale relativa al bonus 50% (se utilizzato), qualora si sia optato per la detrazione diretta.
- **Risparmi, ricavi e costi di gestione annuali**
 - Risparmi da autoconsumo fisico [euro/anno]: risparmi conseguenti all'autoconsumo fisico;
 - Ricavi da energia immessa in rete [euro/anno]: ricavi legati all'energia prodotta dall'impianto fotovoltaico immessa in rete;
 - Totale risparmi e ricavi [euro/anno]: somma delle due voci precedenti;
 - Costi di gestione e manutenzione [euro/anno]: coincide con il dato sui costi di O&M annuali richiesto in input, è riportato tra gli output per confronto con i risparmi e i ricavi.
- **Incentivi e restituzione delle componenti tariffarie annuali**
 - Incentivo MISE sull'energia condivisa [euro/anno]: incentivo erogato dal GSE in base al DM MiSE 16 settembre 2020;
 - Restituzione componenti tariffarie [euro/anno]: in base alla delibera ARERA 318/2020;
 - Restituzione perdite di rete evitate [euro/anno]: in base alla delibera ARERA 318/2020;
 - Totale incentivi e restituzioni [euro/anno]: somma dei campi precedenti.
- **Indicatori finanziari**
 - Tempo di ritorno dell'investimento [anni]
 - Valore attuale netto (VAN) a 20 anni [euro]
 - Tasso interno di rendimento (TIR) [%]
 - Interessi totali sul finanziamento [euro]: nel caso in cui sia presente un finanziamento.

Inoltre, RECON restituisce in un paio di grafici l'analisi energetica su base mensile e l'andamento dei flussi di cassa attualizzati. La Figura 14 riporta l'andamento su base mensile nell'anno 1 di produzione dell'impianto fotovoltaico, autoconsumo diretto, energia condivisa e consumo totale. È quindi possibile

confrontare queste quantità e stimare il margine di produzione fotovoltaica aggiuntiva per aumentare autoconsumo ed energia condivisa. Queste assunzioni possono essere verificate lanciando una nuova simulazione con RECON, aggiornando la taglia selezionata per l'impianto fotovoltaico. La Figura 15 mostra invece l'andamento dei flussi di cassa attualizzati (con il tasso di sconto indicato in input) annuali e cumulati. L'intersezione dei flussi di cassa cumulati con l'asse orizzontale restituisce il tempo di ritorno dell'investimento.

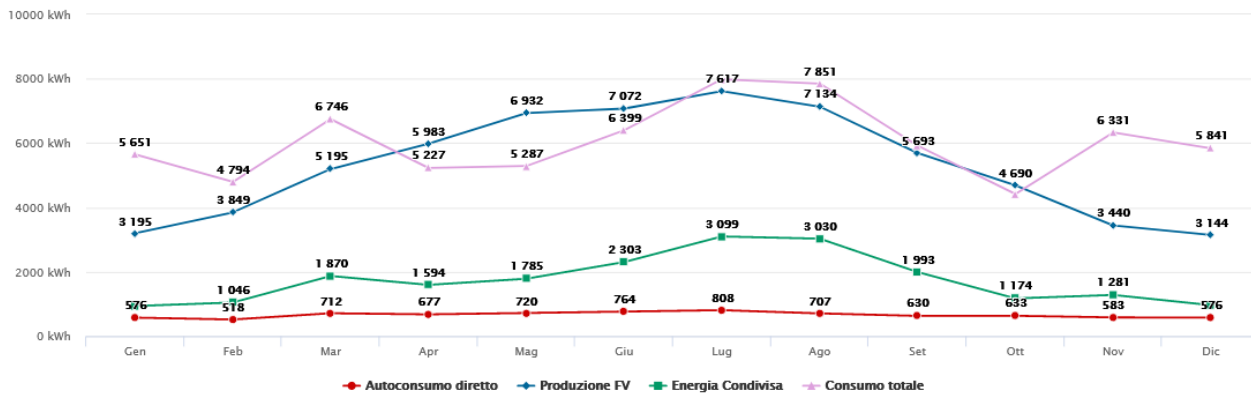


Figura 14. Grafico con l'analisi energetica mensile calcolata da RECON

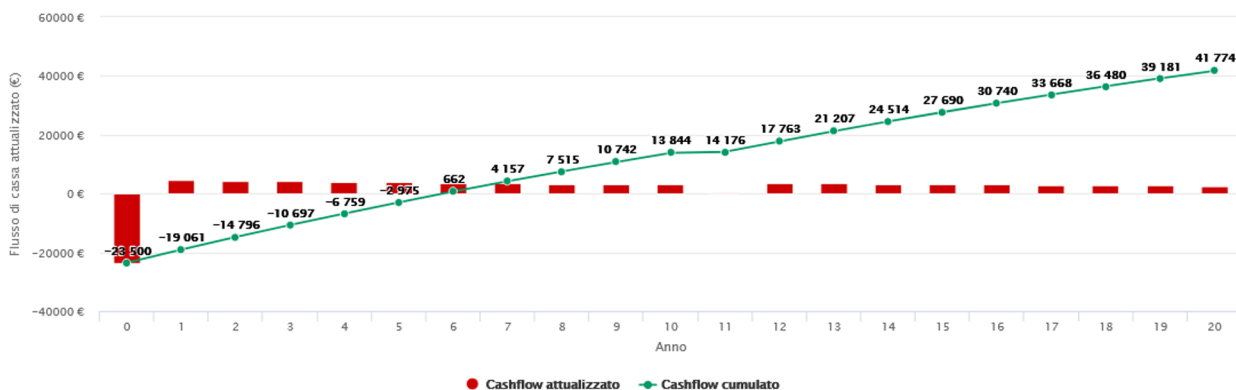


Figura 15. Grafico con i flussi di cassa attualizzati calcolati da RECON

2 Governance della Local Token Economy

Il sistema di Local token Economy prevede di mettere a disposizione dell'amministratore della comunità un ambiente di gestione e controllo "il cruscotto di comunità" nel quale sono integrate alcune funzioni che gli consentono di monitorare e gestire la smart energy community.

Il modello di gestione deve prevedere:

- Parametri di controllo, necessari per regolare adeguatamente il funzionamento della LTE (ad es. valore fee, valore adesione, valore tassazione, fattori di conversione, valore incentivi);
- Indicatori o kpi come strumento di misura qualitativo dell'andamento della LTE (ad es. valore wallet medio, numero transazioni per cittadino, etc);
- un ambiente di simulazione per l'analisi previsionale del bilancio collettivo ed individuale dei token circolanti e quindi prevedere il funzionamento effettivo della Community, basandosi su parametri di ipotesi.

Parametri di controllo

Nello schema seguente si può vedere il flusso dei dati tra simulatore e amministratore LTE

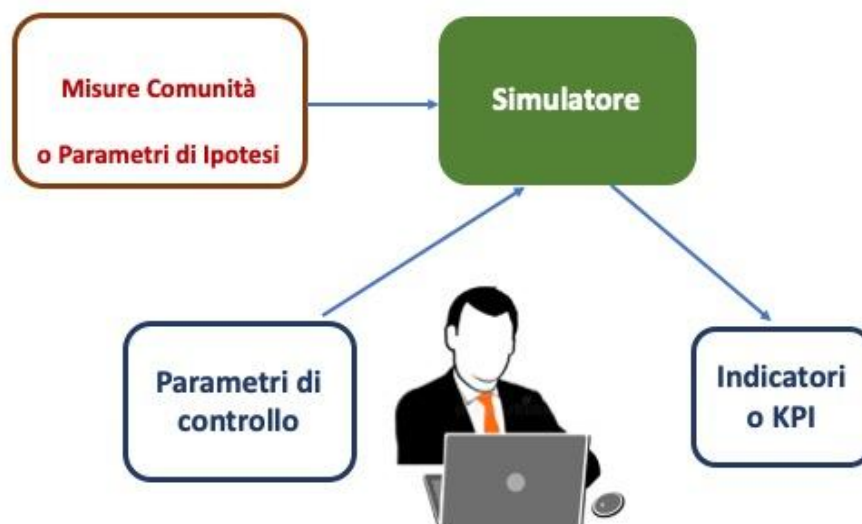


Figura 16: Flusso dei dati in input e output tra simulatore e amministratore LTE

Nel caso in cui non esista una LTE reale il simulatore imposterà come valori che simulano la comunità dei **parametri di ipotesi** mentre nel caso in cui il simulatore sia a effettivo servizio dell'amministratore della LTE i valori saranno acquisiti direttamente dalla community oggetto della simulazione.

I **parametri di controllo** offrono la possibilità di mantenere sotto controllo il mercato dei token della comunità e i loro valori possono variare nel tempo a seconda dell'evolversi delle dinamiche economiche della LTE.

Il mantenimento in equilibrio della comunità e del suo quantitativo di token è uno dei principi chiave della LTE. Per mantenimento in equilibrio della comunità si intende che a regime, ovvero quando tutti i membri sono entrati nella comunità e la *rate* di transazione medio è fisso, il bilancio dei token dovrà essere costante nel tempo senza portare ad una eccessiva presenza o una carenza di token all'interno della piattaforma. Infatti, una delle problematiche che si sono evidenziate è l'elevata divergenza e l'instabilità della piattaforma a seguito di una simulazione con valori non adeguatamente studiati. Questa instabilità porta a divergere su scenari economicamente non sostenibili, dove in un caso si ha una eccessiva presenza di token all'interno della comunità o si va in condizioni di perdita con valori negativi sul bilancio complessivo dei token, scenari questi che risultano essere inaccettabili per il benessere della comunità.

I parametri di controllo identificati e riportati in dettaglio nelle schede a seguire sono:

- Bonus di adesione;

- Tassazione;
- Valore percentuale degli incentivi;
- Demurrage;

Uno dei parametri di controllo più importati è il **bonus di adesione** dei cittadini e degli stakeholder. Infatti, esso rappresenta il principale flusso d'immissione dei token all'interno della comunità e quindi la sua variazione è di vitale importanza per il mantenimento in equilibrio del bilancio dei token a regime. Il Bonus di adesione può essere anche declinato annualmente come Bonus annuale con le medesime caratteristiche. Anche **la tassazione** sulle singole transazioni rappresenta uno dei principali parametri di controllo del bilancio di token all'interno della comunità che deve mantenersi su valori adeguati onde evitare il fenomeno di disincentivazione alla partecipazione alla LTE. Infatti se la tassazione fosse troppo alta i membri sarebbero poco incentivati ad effettuare transazioni, viceversa se fosse troppo bassa ne risentirebbe il bilancio in token per la comunità (si veda Report RdS/PTR(2021)/036).

Un altro importante parametro di controllo è il **valore in percentuale degli incentivi** pagati in token per remunerare le transazioni virtuose dal punto di vista ecosostenibile o socialmente utile. Questi incentivi vengono percepiti sia dall'erogatore che dal fruitore del servizio e possono essere impostati come valore percentuale sul costo del servizio stesso.

Uno strumento aggiuntivo che permette di combattere l'insorgenza di eventuali comportamenti speculativi è la limitazione dell'accumulabilità dei token nei wallet, ovvero il concetto di moneta a demurrage. Infatti, il token può essere accumulato senza alcun limite di tempo all'interno dei wallet; tuttavia, ogni qualvolta venga superato un tetto massimo (controllo sul singolo wallet) i token in eccedenza verranno automaticamente prelevati e reindirizzati in un wallet in gestione all'amministratore di comunità dal quale vengono supportati particolari interventi di natura socio-assistenziale verso la comunità stessa. Da questo punto di vista il principio del decumulo disincentiva il risparmio e fa in modo che i token rientrino in circolazione sotto forma principalmente di consumo, quindi di erogazione di servizi per la comunità.

A seguire una tabella di sintesi nella quale sono raggruppati i parametri di controllo sopra esposti e viene inoltre evidenziato verso quale tipologia di soggetto della comunità quel parametro viene applicato e se risulta essere un parametro fisso o variabile in relazione al prodotto servizio acquistato.

Tabella 2: parametri di controllo per l'amministratore della LTE

Nome parametro	Descrizione parametro	Ruolo	Applicazione	Area di appartenenza		
				Globale	Cittadino	Stakeholder
Bonus-tassa adesione	Il bonus iniziale è il numero di token che vengono corrisposti agli stakeholder o ai cittadini all'ingresso nella piattaforma	Vengono utilizzati per permettere ad un cittadino e/o stakeholder di poter effettuare delle transazioni al momento dell'adesione	specifico per categoria di utente			
Bonus-tassa annuale	Il bonus annuo indica il bonus che viene corrisposto agli stakeholder o ai cittadini per ogni anno di partecipazione alla piattaforma	Viene utilizzato per far sì che gli stakeholder e/o i cittadini possano ricevere token attraverso il versamento di una quota associativa e gli venga riconosciuto un corrispettivo in token. Questo parametro è importante soprattutto per gli stakeholder che potrebbero utilizzare i token per incentivare i cittadini ad usufruire dei propri servizi senza però avere la possibilità di mettersi a disposizione della comunità	specifico per categoria di utente			
Valore della tassazione	Il valore della tassazione sulle transazioni indica i token ritirati per ogni transazione; può essere applicato in percentuale sul valore della transazione	Ha lo scopo di definire la quantità di token che vengono tassati per ogni transazione	globale			
Incentivo verso erogatore dell'azione	indica in token quando un'azione è virtuosa cioè ecosostenibile o socialmente utile.	Serve a variare il guadagno suggerito a seconda di quanto un'azione sia virtuosa	specifico x categoria/azione			
Incentivo verso fruitore dell'azione	indica in token quando un'azione è virtuosa cioè ecosostenibile o socialmente utile.	Serve a variare il prezzo suggerito a seconda di quanto un'azione sia virtuosa	specifico x categoria/azione			
Prezzo mutualistico orario per i servizi tra membri della comunità	Il prezzo mutualistico è un prezzo orario, uguale per qualsiasi tipo d'azione	Il ruolo del prezzo mutualistico orario è quello di permettere alle persone meno abbienti di poter ricevere dei servizi a cui non potrebbero accedere se dovessero pagare il prezzo di mercato	globale			
Prezzo mutualistico orario per i servizi di comunità	Il prezzo mutualistico applicato ai servizi di comunità pagato dalla comunità che vengono svolti dai membri della comunità	La remunerazione delle attività può essere utilizzata per bilanciare il sistema economico in token della comunità (anche a sostegno dei più bisognosi), quindi più alto del prezzo mutualistico orario per servizi tra membri	globale			
Demurrage	Indica il valore di soglia oltre il quale non è possibile accumulare token	Serve ad incentivare lo scambio e l'uso del token e ad aumentare l'utilizzo dei servizi di comunità. I token in eccedenza vengono accumulati nel wallet di solidarietà	globale			

Per alcuni parametri di controllo è stata realizzata una scheda descrittiva, di seguito riportata.

Nome Parametro di Controllo:
Bonus di Adesione
Obiettivo:
Fornire agli utenti della comunità un mezzo per poter subito essere attivi al suo interno. Configura alcuni aspetti del funzionamento delle comunità
Descrizione:
Il bonus iniziale è il numero di token che vengono corrisposti agli stakeholder o ai cittadini all'ingresso nella piattaforma.
Unità di Misura:
Token
A cosa serve
Il bonus di adesione in token permette ad un cittadino e/o stakeholder di poter effettuare delle transazioni già al momento dell'adesione.
ATTIVAZIONE
Viene attivato e impostato dall'amministratore della token economy. Per il cittadino: al momento di adesione alla LTE (ed a rinnovo annuale) Per lo stakeholder: all'atto del versamento della quota di iscrizione alla LTE (ed a rinnovo annuale)

Nome Parametro di Controllo:
Valore della Tassazione
Obiettivo:
Configura alcuni aspetti del funzionamento delle comunità. Serve a far rientrare nella comunità un certo numero di token e controllarne il flusso.
Descrizione:
Il valore della tassazione indica i token ritirati sul valore della transazione ed è espressa in percentuale sul costo dell'azione. Viene applicata al fruitore.
Unità di Misura:
in Token, ad es. pari al 10% sul prezzo esposto su marketplace

A cosa serve
Ha lo scopo di definire la quantità di token che vengono tassati per ogni transazione, utili a sostenere la comunità.
ATTIVAZIONE
Viene attivata e regolata dall'amministratore della token economy. Per cittadino/stakeholder fruitore: al momento della transazione (in passivo). Per la comunità: al momento della transazione (in attivo).
Esempio
Acquisto bene usato (bicicletta) - prezzo in token 30 Token su marketplace. Maria vende la propria bicicletta usata a Francesco che spende -30 - 3 (10% tassazione) = - 33 token mentre Maria guadagna +30 token; per il wallet di comunità si ha un bilancio positivo di +3 token (tassazione).

Nome Parametro di Controllo:
Incentivo ambientale o sociale
Obiettivo:
Incentivare azioni sostenibili dal punto di vista ambientale e sociale
Descrizione:
Indica quanto un'azione è socialmente / ambientalmente rilevante per la comunità e quindi rappresenta un extra profitto per i soggetti coinvolti nell'azione ed è differenziato tra erogatore e fruitore.
Unità di Misura e valore:
Token: l'incentivo è calcolato in percentuale sul prezzo dell'azione ed è ripartito per valori % separati tra fruitore ed erogatore (c'è un valore soglia di controllo per la somma delle due %); l'incentivo è in aggiunta al prezzo della azione ed è pagato dalla comunità. Il prezzo orario del sistema mutualistico è uguale per tutti i servizi perché si dà valore al tempo e si lascia facoltà al gestore di fissare la percentuale d'incentivo in relazione alla valenza sociale/ambientale del servizio.
A cosa serve
Serve a valorizzare le azioni di sostenibilità sociale ed ambientale.
ATTIVAZIONE
Attivato e gestito dall'amministratore della token economy Per cittadino/stakeholder: al momento della transazione (in attivo). Per la comunità: al momento della transazione (in passivo).
Esempio
Servizio di babysitteraggio - prezzo in token 10 T/h su marketplace. Se ipotizziamo un incentivo massimo del 25% pari a 2,5 token sia per fruitore che per erogatore, la mamma spende -10 + 2,5 (incentivo) - 1 (tasse) = - 8,5 token mentre la babysitter guadagna 10 +2,5 = +12,5 token e per il wallet di comunità si ha un bilancio di -5 (incentivo) +1 (tassazione) = - 4 token.

Nome Parametro di Controllo:
Demurrage
Obiettivo:
incentivare lo scambio e l'uso del token e ad aumentare l'utilizzo dei servizi di comunità.
Descrizione:
Il demurrage è un valore soglia oltre il quale non è possibile accumulare token all'interno del wallet di un membro della comunità. Il controllo va fatto ad ogni transazione e deve esserci un avviso che il wallet ha un limite oltre il quale non è possibile cumulare. I token in eccedenza vengono stornati nel wallet di solidarietà.
Unità di Misura:
Token
A cosa serve
Serve a limitare l'accumulo di token da parte di uno/più membri della comunità e nel contempo può finanziare azioni di sostegno per i membri più fragili della comunità (Vedi povertà energetica).
ATTIVAZIONE
Viene attivato al raggiungimento del valore soglia impostato dall'amministratore della token economy.
Esempio
Si ipotizza che Alice abbia 196 token nel proprio wallet e che eroghi un servizio di babysitteraggio e guadagni 10 token. Il wallet si incrementa fino a 200 token (+4 token) perchè i +6 token in eccedenza andranno caricati direttamente nel wallet di solidarietà.

Gli indicatori

Un altro strumento a disposizione dell'amministratore della LTE è rappresentato da un set di **indicatori o kpi** opportunamente scelti e che permettono di definire in modo univoco lo status dell'economia dei token della comunità e il suo equilibrio.

I KPI sono raggruppati in 3 tipologie:

- Indicatori economici;
- indicatori energetico – ambientali;
- indicatori sociali.

Le tabelle seguenti riportano l'elenco degli indicatori presenti a disposizione dell'amministratore della LTE.

Tabella 3: Indicatori economici

Nome Indicatore	Descrizione Indicatore	Ruolo	Unità di misura
PIL (Prodotto interno lordo)	Il PIL è l'ammontare del valore dei beni e servizi scambiati nel mese considerato ed è costituito dal valore esatto (costo dell'azione) più le tasse nette pagate (aggiunte in quanto componenti del prezzo fino al pagamento dagli acquirenti)	Serve per valutare la ricchezza in termini di beni e servizi ben scambiati. Quindi il PIL può essere considerato come la spesa totale della comunità in token nel mese considerato e tale valore al fine del controllo dovrà sempre aumentare e tendere ad un valore all'infinito.	Token
PIL pro capite	Il PIL pro capite è l'ammontare del valore dei beni e servizi scambiati nel mese considerato per il singolo membro ed è il rapporto tra il PIL e il numero totale dei membri della comunità	Serve per valutare la ricchezza in termini di beni e servizi ben scambiati per membro e quindi la spesa sostenuta. Tale valore al fine del controllo dovrà sempre aumentare e tendere ad un valore limite.	Token/membro
GC (guadagno comunitario)	GC è la somma dei redditi in token percepiti dai membri della comunità in un certo lasso di tempo (mensile). Tali redditi possono essere: compensi per servizi ben ceduti e incentivi. GC è costituito dall'ammontare dei token scambiati che rappresenta il valore totale dei beni e servizi scambiati (token che ricevono gli erogatori) più gli incentivi pagati dalla comunità agli erogatori stessi.	Quanto indica il guadagno lordo in token della comunità e come per il PIL tale valore deve sempre crescere e tendere all'infinito ad un valore limite ottimale.	Token
GC pro capite	GC pro capite è il rapporto tra il guadagno comunitario e il numero dei suoi membri.	Quanto indica il guadagno medio lordo in token per ogni bitante nel mese considerato.	Token/membro
RC (reddito comunitario)	Il reddito comunitario è la differenza tra il GC e il PIL e quindi rappresenta il reddito netto tra le entrate e le uscite della comunità nel mese considerato.	Quanto indica il reddito netto lordo in token della comunità nel mese considerato.	Token
RC pro capite	Il reddito comunitario pro capite è la differenza tra il GC e il PIL diviso per il numero totale dei membri della comunità e quindi rappresenta il reddito netto tra le entrate e le uscite della comunità nel mese considerato.	Quanto indica il reddito netto lordo in token per ogni bitante nel mese considerato.	Token/membro
Capitale Proprio	Il capitale proprio è l'ammontare dei token che possono essere spesi/investiti dagli stakeholder o cittadini nel primo mese ed è la somma dell'ammontare dei token residui del mese precedente e i token immessi nel sistema costituiti dai token per il bonus di adesione e token a quota (solo nel caso degli stakeholder).	Tale valore quanto indica la disponibilità propria in token dei cittadini e degli stakeholder nel primo mese e più grande più i cittadini/stakeholder possono spendere. Tale valore non deve essere mai negativo in quanto si è ipotizzato che i wallet non passano mai in debito.	Token
Capitale Totale	Il capitale totale è la somma del capitale proprio degli stakeholder/cittadini nel mese primo più gli incentivi erogati dal sistema che possono essere spesi/investiti nel primo mese.	Il capitale totale quanto indica la disponibilità totale in token dei cittadini e degli stakeholder nel primo mese e questo valore non deve essere mai negativo in quanto si è ipotizzato che i wallet non passano mai in debito.	Token
Wallet medio cittadino	Il wallet medio cittadino è la disponibilità media a fine mese dei cittadini ed è il rapporto tra la quantità totale di token dei cittadini e il numero di quest'ultimi.	Ha il compito di determinare la politica di demerage che deve essere eseguita durante la fase di controllo in quanto il wallet medio si dovrà assestare su un valore fisso.	Token/membro
Wallet medio stakeholder	Il wallet medio stakeholder è la disponibilità media a fine mese dei stakeholder ed è il rapporto tra la quantità totale di token dei stakeholder presenti e il numero di quest'ultimi.	Ha il compito di determinare la politica di demerage che deve essere eseguita durante la fase di controllo in quanto il wallet medio si dovrà assestare su un valore fisso.	Token/membro
Indice di occupazione oraria	L'indice di occupazione oraria all'interno della comunità è un rapporto che ci permette di definire quante ore vengono spese attraverso i servizi della comunità dai membri rispetto a tutte le ore che potrebbero essere spese dall'intera community.	Questo indice ha un puro ruolo informativo dello status della comunità e può rappresentare in un certo modo il grado di partecipazione. Infatti, più questo valore è grande più la comunità è attiva e scambia servizi e partecipa alla vita della comunità.	adimensionale [%]
IAE (Indice di attività dell'economia)	L'indice di attività dell'economia è il rapporto tra i token scambiati da tutti i membri nel mese primo rispetto a un numero di token presenti sempre nel mese stesso.	Questo indice confronta i token scambiati con quelli presenti nella LTE; se il valore dell'IAE è minore di 1 significa che solo una piccola parte dei token presenti ha non effettivamente circolato nella nostra comunità (quindi una comunità poco dinamica). Invece, se il valore è maggiore di 1 sicuramente una buona parte dei token presenti ha circolato anche più volte (comunità molto dinamica).	adimensionale [%]

Tabella 4: Indicatori energetico-ambientali

Nome Indicatore	Descrizione Indicatore	Ruolo	Unità di misura
Indice di indipendenza energetica	Il fattore di indipendenza energetica è il rapporto tra l'energia autoprodotta e il fabbisogno totale energetico della comunità. Questo fattore quindi è il rapporto tra l'energia totale prodotta e l'energia totale richiesta nel periodo di calcolo considerato.	Ha il compito di determinare l'indipendenza energetica della comunità dalla rete e tale indice deve essere sempre almeno uguale ad un valore di soglia. Tale valore può essere anche maggiore di 1 in tal caso sarà presente un valore di energia esula al mercato.	adimensionale [%]
Indice di autosufficienza	È il rapporto tra l'energia autoconsumata e il totale dell'energia autoprodotta dagli impianti presenti nella comunità.	È uno degli indici più importanti in quanto questo valore deve essere sempre pari almeno al valore 1, perciò la componente autoprodotta deve essere sempre valorizzata consumandola interamente.	adimensionale [%]
Incentivi ambientali pro capite	È il totale degli incentivi ambientali rilasciati dalla comunità mediamente verso il singolo membro nel mese considerato.	Tale indice serve per definire mediamente quanto un membro riceve la componente ambientale di incentivazione.	Token/membro

Tabella 5: Indicatori sociali

Nome Indicatore	Descrizione Indicatore	Ruolo	Unità di misura
Indice di socialità	L'indice di socialità è un indice che rapporta il numero totale di azioni a sfondo sociale sul numero totale di transazioni (quindi di azioni) eseguite dai membri della comunità fino al periodo della simulazione.	L'indice di socialità serve a determinare se incentivare o meno un set determinato di attività sociali.	adimensionale [%]
Indice di risparmio economico pro capite	Indice di risparmio economico è il totale ammontare degli euro risparmiati mediamente dal singolo cittadino nel mese considerato relativi al soddisfacimento dei propri bisogni attraverso transazioni mutualistiche.	Il ruolo di questo indice è di valutare il risparmio economico della comunità così che la comunità può investire il capitale risparmiato per altri bisogni. Il valore di tale indice dovrà essere proporzionato alle transazioni effettuate.	€/membro
Rapporto di attività	Il rapporto di attività è un indice che indica il numero di membri attivi su i membri totali presenti ed è ottenuto attraverso il rapporto dei wallet/utenti attivi sul numero totale dei membri della nostra comunità. Questo indice è il complementare del rapporto di inattività.	Questo indice ha il compito di indicare il peso degli attivi e durante il controllo si cercherà di mantenere tale valore più alto possibile.	adimensionale [%]
Rapporto di inattività	Il rapporto di inattività è un indice che indica il numero di membri inattivi sui membri totali presenti ed è ottenuto attraverso il rapporto dei wallet/utenti inattivi sul numero totale dei membri della nostra comunità.	Questo indice ha il compito di indicare il peso degli inattivi sui totali presenti e durante il controllo si cercherà di mantenere tale valore più basso possibile.	adimensionale [%]

2.1 Il modello ad agenti Smart Token Community

Nella precedente LA è stato implementato un simulatore in ambiente Matlab (si veda Report RdS/PTR(2021)/039) avente un approccio statistico su base globale per l'intera comunità che non prevede un sistema di controllo a livello di singolo membro. L'impossibilità di associare le singole transazioni ai wallet dei singoli membri fa sì che i token scambiati non siano legati alla disponibilità del wallet del singolo membro. Per superare questi limiti e simulare le dinamiche di evoluzione di una comunità con scambio di beni e servizi basato su token oltre a prevedere la dinamica di adesione di nuovi membri, è stato implementato un modello ad agenti in ambiente Python. Gli agenti sono i cittadini e gli stakeholder che aderiscono o possono aderire alla comunità. Il codice di calcolo implementato si basa su alcune teorie comportamentali tipicamente utilizzate nella modellazione ad agenti, in particolare la Teoria del comportamento previsto (*Theory of planned behavior*) e il modello dell'Accordo relativo (*Relative agreement*).

Nelle sezioni che seguono sono riportati l'indagine preliminare condotta per la sezione delle teorie comportamentali implementate nel modello ad agenti, la descrizione delle teorie selezionate e le caratteristiche del modello implementato.

2.1.1 Indagine preliminare sulle teorie comportamentali per il modello ad agenti

Inizialmente è stata condotta un'indagine in letteratura su alcune teorie comportamentali, propedeutica alla selezione del modello da implementare. Le teorie prese in considerazione sono:

- Utility function
- Theory of planned behavior
- Logistic regression

Di seguito sono sinteticamente riportati i pro e i contro rilevati.

Utility function è una funzione lineare pesata caratteristica di ogni agente, in cui la somma dei pesi è unitaria, i fattori sono normalizzati e assumono valori tra 0 e 1. Il valore di tale funzione viene confrontato con una soglia per stabilire se l'agente prende una determinata decisione. I fattori, o *partial utilities*, sono scelti a discrezione dello sviluppatore, così come il loro calcolo e la loro normalizzazione.

Pro: possono essere presi in considerazione svariati aspetti nel processo decisionale, lo sviluppatore può dare un peso diverso ai vari fattori calibrato sullo specifico caso studio.

Contro: complessità nel selezionare quali fattori considerare, i fattori vanno calcolati e normalizzati con specifiche equazioni. Sono richieste specifiche assunzioni in fase di modellazione (linearità, normalizzazione, indipendenza).

Theory of planned behavior (TPB): è una teoria comportamentale particolarmente versatile, in base alla quale il comportamento umano può essere descritto principalmente da tre attributi: attitudine verso il comportamento, influenza da parte degli altri agenti che fanno parte del network sociale, e controllo percepito sui propri comportamenti.

Pro: modello dettagliato e flessibile, è possibile selezionare i vari attributi comportamentali applicabili al caso studio.

Contro: per ottenere risultati affidabili è necessario definire adeguatamente i pesi delle relative funzioni oltre ai fattori che influenzano gli attributi comportamentali. È necessario implementare molti attributi per simulare appropriatamente il comportamento dei cittadini.

Logistic regression: stima la probabilità che un evento accada sulla base di variabili indipendenti. Le regole comportamentali riguardo una specifica azione sono definite come funzioni probabilistiche che esprimono quanto è probabile che quell'azione avvenga, basandosi sui diversi fattori.

Pro: a differenza della *Utility function* non richiede assunzioni iniziali.

Contro: richiede che le variabili utilizzate nella funzione di probabilità siano tra loro indipendenti, il che richiede un'apposita verifica prima di procedere con la simulazione. I coefficienti di regressione devono essere ricavati sperimentalmente attraverso una procedura di fitting. La funzione di probabilità è più complessa delle funzioni lineari, con conseguenti maggiori costi computazionali.

Sulla base dell'indagine condotta e tenendo conto delle finalità del modello da implementare, si è deciso di utilizzare la Teoria del comportamento previsto.

2.1.2 La teoria del comportamento previsto (*theory of planned behavior*)

La Teoria del comportamento previsto (*Theory of Planned Behavior* - TPB) è una teoria comportamentale sviluppata da Ajzen [1] - [2] che trova largo impiego nei modelli ad agenti grazie alla sua versatilità ed efficacia. Tre sono i principali fattori da considerare nell'applicazione di tale teoria:

- *Behavioral beliefs*: sono le convinzioni personali di ottenere un certo risultato da uno specifico comportamento. Questo fattore influenza direttamente la *Attitude toward the behavior* (*att*);
- *Normative beliefs*: rappresentano l'influenza generale su di un agente da parte dagli agenti che costituiscono il suo network sociale. È la percezione di un individuo rispetto a una certa azione che è significativamente influenzata dal giudizio di altre persone che esercitano un'influenza su quell'individuo. Esso compone il termine *Subjective norm* (*sn*);
- *Control beliefs*: è la percezione che un individuo ha della sua capacità e possibilità, in termini di mezzi e risorse, di compiere una determinata azione. Esso influenza il termine *Perceived behavioral control* (*pcb*).

Prendendo spunto da [3] questi fattori sono stati implementati come segue. La grandezza *Attitude toward the behavior* è funzione dell'opinione generale dell'agente, che evolve nel tempo:

$$att = \frac{opi+1}{2} \quad \text{Eq. 1}$$

La grandezza *Subjective norm* (*sn*) è la combinazione di *Social pressure* (*sp*) e *Innovativeness* (*innov*). La *Social pressure* è calcolata considerando l'influenza generale che un agente riceve dal proprio network sociale; vengono considerate in questo calcolo solo le opinioni forti, caratterizzate da un'opinione $opi > 0,6$ e da un'incertezza dell'opinione $unc < 0,5$. In seguito alla selezione di N agenti che esercitano influenza sull'agente considerato, prima si calcola la pressione sociale che l'agente j esercita sull'agente i ; infine, la *Social pressure* (*sp*) è il valore medio tra tutte quelle considerate:

$$w_{unc} = \frac{\frac{unc_j}{unc_i} - 0,2}{20 - 0,2} \quad \text{Eq. 2}$$

$$sp_{ij} = att_j(1 - w_{unc}(att_j - att_i)) \quad \text{Eq. 3}$$

$$sp = \frac{\sum_{j=0}^N sp_{ij}}{N} \quad \text{Eq. 4}$$

L'*Innovativeness* (*innov*) è il livello di innovazione che caratterizza ogni agente. Questo attributo influenza il termine *Subjective norm* (*sn*) e segue la teoria sviluppata da Rogers [4]. In base a questa teoria, gli agenti possono essere suddivisi in cinque categorie con diversi livelli di propensione all'innovazione: innovatori (percentuale sulla popolazione pari a 2,5%), utilizzatori iniziali (13,5%), maggioranza della prima ora (34%), maggioranza ritardataria (34%), ritardatari (16%).

L'*Innovation factor* (*innov_f*) è il livello di innovazione normalizzato con la seguente formula:

$$innov_f = \frac{innov-1}{4} \quad \text{Eq. 5}$$

Infine, la *Subjective norm* (*sn*) è calcolata come combinazione di *Social pressure* e *Innovation factor*:

$$sn = sp * w_{sp} + innov_f * w_{innov_f} \quad \text{Eq. 6}$$

dove i rispettivi pesi w sono assunti pari a 0,5 (stessa importanza), quindi il risultato è una media tra i due fattori. La calibrazione dei suddetti pesi andrebbe effettuata sulla base dello specifico caso studio reale.

Nel modello ad agenti implementato, la grandezza *Perceived behavioral control* (*pcb*) dipende unicamente dal numero di ore settimanali spese dal singolo agente (cittadino) per erogare/usufruire dei servizi erogati ai cittadini all'interno della comunità. Nel caso in cui vi siano due tipologie di servizi attivati nella comunità, il *pcb* sarà calcolato con la seguente relazione:

$$pcb = \frac{(ore_{s1} + ore_{s2}) - 1}{8} \quad \text{Eq. 7}$$

La normalizzazione deve essere aggiornata ogni volta che sono aggiunti servizi o nel caso in cui sia modificato l'intervallo di disponibilità oraria ore_{si} del servizio i -esimo.

I termini sono quindi rielaborati per comporre l'intenzione del comportamento (*behavioral intention*, *bi*):

$$bi = att * w_{att} + sn * w_{sn} + pcb * w_{pcb} \quad \text{Eq. 8}$$

dove i pesi (w) sono definiti a partire dall'analisi delle statistiche dei gruppi sociali (considerando fattori socio-demografici).

Il comportamento finale (*behavior*, b) è calcolato come combinazione tra bi e pcb :

$$b = bi * w_{bi} + pcb * w_{pcb} \quad \text{Eq. 9}$$

$$w_{pcb} = 1 - w_{bi} \quad \text{Eq. 10}$$

dove w_{bi} è un parametro del modello (generalmente maggiore di 0,5).

Il comportamento finale è un numero compreso nell'intervallo $[0, 1]$ ed è confrontato con un valore di soglia b_{thsd} . Nel caso in cui b superi il valore di soglia, l'agente prende la decisione relativa al comportamento analizzato (ingresso nella token community), e di conseguenza gli viene assegnata un'opinione $opi = 0,8$ e un'incertezza dell'opinione $unc = 0,1$, ovvero un'opinione forte a aderire alla comunità. Questi parametri fanno parte del modello del *Relative Agreement*, che è utilizzato nel modello ad agenti implementato ed i cui dettagli sono riportati nella sezione successiva.

2.1.3 La teoria dell'accordo relativo (*relative agreement*)

Il modello del *Relative Agreement* (RA) è usato per modellare la dinamica dello scambio di opinioni tra gli agenti, sulla base della teoria sviluppata da Deffuant [5]. Nel RA coppie di agenti interagendo cambiano la loro opinione; ogni agente i interagisce casualmente con un agente j del proprio network sociale e lo influenza. Ogni agente è caratterizzato da un'opinione e da una corrispondente incertezza che, in termini matematici, possono essere rappresentati tramite segmenti, come riportato in Figura 1:

$$s_i = [opi_i - unc_i; opi_i + unc_i] \quad \text{Eq. 11}$$

$$s_j = [opi_j - unc_j; opi_j + unc_j] \quad \text{Eq. 12}$$

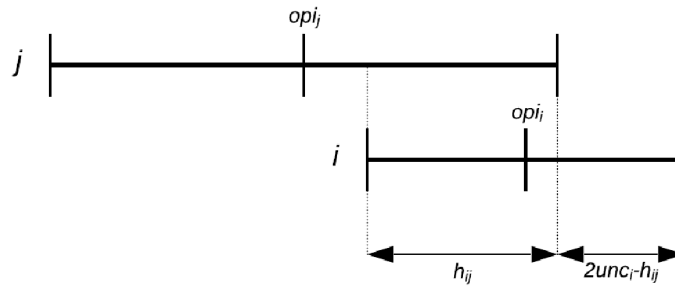


Figura 1. Schematizzazione della teoria del Relative agreement

L'interazione determina la sovrapposizione di opinione h_{ij} , per cui un agente viene influenzato solamente da altri agenti che hanno un'opinione simile alla propria. Tale sovrapposizione è data dalla seguente equazione:

$$h_{ij} = ((opi_i + unc_i; opi_j + unc_j)) - \max((opi_i - unc_i; opi_i + unc_i)) \quad \text{Eq. 13}$$

L'accordo (*agreement*) è calcolato come:

$$h_{ij} - (2unc_i - h_{ij}) = 2(h_{ij} - unc_i) \quad \text{Eq. 14}$$

mentre l'accordo relativo (*relative agreement*) è:

$$\frac{2(h_{ij} - unc_i)}{2unc_i} = \frac{h_{ij}}{unc_i} - 1 \quad \text{Eq. 15}$$

Qualora $h_{ij} \leq unc_i$, allora i non influenza j ; viceversa, se $h_{ij} > unc_i$, l'opinione e l'incertezza dell'agente j sono modificate in base alle seguenti relazioni:

$$opi_j = opi_j + \mu \left(\frac{h_{ij}}{unc_i} - 1 \right) (opi_i - opi_j) \quad \text{Eq. 16}$$

$$unc_j = unc_j + \mu \left(\frac{h_{ij}}{unc_i} - 1 \right) (unc_i - unc_j) \quad \text{Eq. 17}$$

Dove μ è un parametro costante che controlla la velocità di convergenza delle opinioni.

2.1.4 La struttura del modello ad agenti

In questa sezione sono descritti la struttura e il funzionamento del modello ad agenti implementato in ambiente Python per simulare l'evoluzione di una 'Smart token community' basata sull'uso di token per lo

scambio di beni e servizi. Il modello è in grado di simulare l'andamento degli scambi, in termini di token scambiati da ciascun membro della comunità e dalla comunità stessa, l'andamento degli introiti di valuta a corso legale (euro) in ingresso nella comunità a seguito dell'adesione di nuovi membri (fee di adesione), e il numero di adesioni di nuovi membri nel tempo. Il modello implementato è quindi uno strumento di simulazione in grado di stimare l'evoluzione di una comunità basata su token, al fine di valutare l'impatto di alcuni parametri (es. fee per singola transazione, bonus adesione, numero di servizi erogati nella comunità) e quindi adottare le opportune strategie e policy.

Il modello è scritto in linguaggio Python, e, oltre alle comuni librerie open-source, tra cui MATH, RANDOM, NUMPY, PANDAS, si avvale di MESA, una libreria dedicata allo sviluppo di modelli ad agenti [6]. In particolare, MESA mette a disposizione un'interfaccia grafica da cui è possibile impostare i parametri *user settable*, ovvero parametri il cui valore può essere impostato dall'utente nella GUI del frontend (all'interno dei limiti e con lo step definiti nel codice sorgente) in fase di inizializzazione prima di avviare la simulazione. La stessa GUI riporta anche una dashboard con le principali grandezze che si aggiornano in tempo reale all'avanzare della simulazione, attraverso una serie di grafici che saranno descritti nel seguito. Un estratto della GUI è riportato in Figura 2.

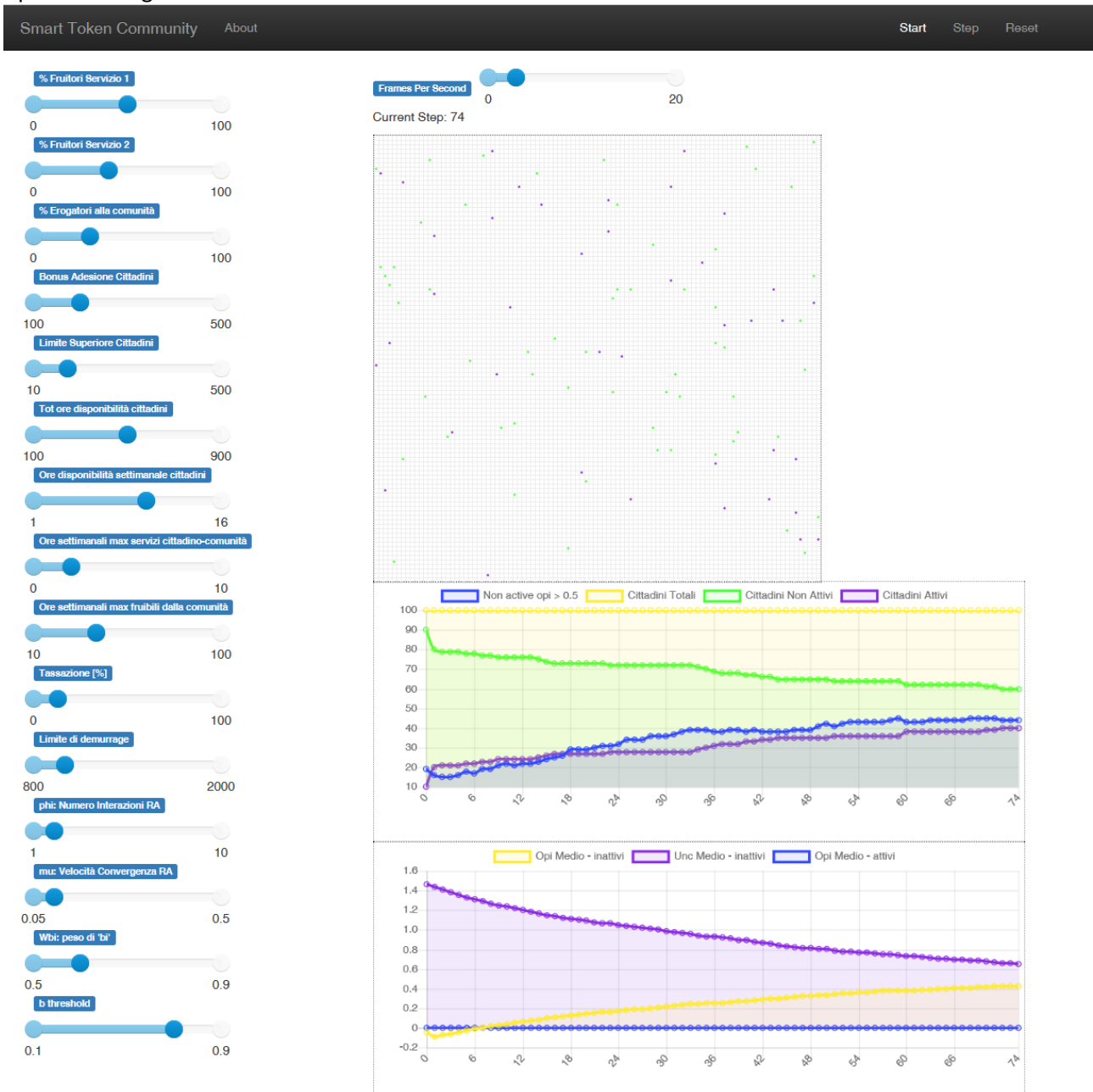


Figura 17: Estratto della GUI del modello ad agenti “Smart Token Economy”

2.1.5 I parametri del modello

L’attuale versione del codice considera lo scambio di servizi intesi come ore che possono essere dedicate dai cittadini per erogare un determinato servizio ad altri cittadini (fruitori) e alla comunità.

I parametri sono impostati dall’utente attraverso l’interfaccia grafica di MESA in fase di inizializzazione del modello stesso. Con riferimento alla Figura 17 si riportano di seguito i parametri impostabili dall’utente e la relativa spiegazione.

- *%Fruitori Servizio i*: per ciascuno dei servizi scambiati all’interno della comunità, rappresenta la percentuale iniziale di fruitori, rispetto al totale dei cittadini, per un dato servizio i-esimo. Nell’attuale versione, il complementare a cento della popolazione iniziale è identificata come erogatore. Tale percentuale varia nel corso della simulazione con l’ingresso di nuovi cittadini nella comunità, in quanto i nuovi membri vengono inizializzati randomicamente come erogatori/fruitori tramite la funzione ‘*initialize_new_citizens*’.
- *%Erogatori alla comunità*: percentuale iniziale di cittadini che erogano servizi a favore della comunità. Questo valore varia con l’ingresso di nuovi cittadini nella comunità. Tale parametro viene trattato esattamente come il precedente nelle funzioni ‘*initialize_users_providers*’ e ‘*initialize_new_citizens*’, ma identifica una tipologia diversa di servizio.
- *Bonus Adesione Cittadini [token]*: ogni cittadino riceverà una quantità di token pari al valore impostato quando entrerà nella comunità.
- *Limite Superiore Cittadini*: indica la popolazione massima di agenti.
- *Tot ore disponibilità cittadini*: numero massimo di ore che un cittadino può dedicare - come erogatore o fruitore – per servizi resi al cittadino durante l’intera simulazione, esclusi i servizi resi alla comunità.
- *Ore disponibilità settimanale cittadini*: numero massimo di ore dedicate (erogate o fruite) da un cittadino durante una settimana. L’intervallo temporale di riferimento è attualmente posto pari alla settimana ma può essere variato tramite un apposito parametro agendo sul codice sorgente.
- *Ore settimanali max servizi cittadino-comunità*: numero massimo di ore che un cittadino può erogare in favore della comunità, durante l’intervallo temporale definito (es: settimana).
- *Ore settimanali max fruibili dalla comunità*: numero massimo di ore che la comunità può complessivamente ricevere da parte dei cittadini nell’unità temporale definita, coincidente con la settimana.
- *Limite di demurrage*: quantità di token, contenuti nel wallet personale di ciascun agente, oltre alla quale viene applicato il meccanismo di demurrage.
- *Tassazione [%]*: percentuale di token che vengono prelevati dalla comunità ad ogni transazione sotto forma di fee. I servizi che i cittadini erogano in favore della comunità non sono soggetti a fee.
- *phi: Numero Interazioni RA*: rappresenta il numero di interazioni che avvengono tra cittadini secondo l’algoritmo RA descritto nel seguito (parametro ϕ).
- *mu: Velocità Convergenza RA*: esprime la velocità di convergenza delle opinioni secondo la teoria della dinamica di opinione dell’algoritmo RA (parametro μ).
- *Wbi: peso di ‘bi’*: rappresenta il peso del termine *Behavioral intention* (bi) della Teoria del comportamento previsto.
- *b threshold*: rappresenta il valore di soglia superata la quale un agente entra a far parte della comunità, da confrontare con il termine *Behavior* (b) della Teoria del comportamento previsto.

2.1.6 Le classi del modello

2.1.6.1 Scheduler

È la classe del modello che gestisce il tempo. Lo scheduler eredita le proprietà della classe ‘*StagedActivation*’ disponibile nel pacchetto MESA. Ogni step temporale viene suddiviso dallo scheduler in più stadi, e ad ogni stadio corrispondono determinate azioni definite nel modello. Ogni agente viene attivato in ordine random

in un dato stadio e tutti gli agenti devono essere attivati in quello stadio prima di passare a quello successivo. Allo stato attuale gli stadi sono due: il primo resetta i contatori ad ogni scadenza temporale mentre il secondo attiva le funzioni di erogazione/fruizione servizi nella comunità.

2.1.6.2 Community

La classe *community* è una sottoclasse della classe 'Agent' di MESA ed è implementata per considerare tutti quei comportamenti e fenomeni all'interno della 'Smart token community' che prevedono la regolazione e supervisione della comunità stessa.

Gli attributi legati alla comunità sono creati innanzitutto per tenere conto dei flussi di cassa sia in euro sia in token della comunità. Inoltre, la classe contiene la funzione *community_balance* che calcola il bilancio globale dei flussi di token della comunità.

2.1.6.3 Citizens

La classe dei cittadini è la più ampia del modello, in quanto contiene gli attributi e le funzioni che permettono di gestire i cittadini e le loro interazioni a livello di scambio di servizi nella comunità.

Gli attributi della classe cittadini riguardano:

- i parametri legati all'algoritmo del RA (opinione e incertezza dell'opinione);
- l'adesione del singolo cittadino alla comunità, in funzione del valore che assume l'attributo *behavior* in accordo con la teoria TPB;
- il portafoglio virtuale (wallet) del singolo agente nella comunità, tiene traccia di tutte le transazioni effettuate dal singolo agente;
- i contatori delle transazioni di ogni singolo agente con altri agenti (servizi tra cittadini) e con la comunità (servizi dei cittadini nei confronti della comunità);
- le fee che vengono trattenute ad ogni transazione al singolo agente. La somma delle fee tutti gli agenti viene poi raccolta nel corrispondente attributo della comunità;
- attributo *community* che serve a legare la classe della comunità ad ogni agente;
- attributo *model* che serve a legare la classe del modello ad ogni agente;
- attributo che identifica se un agente è fruitore o erogatore di un determinato servizio;
- attributo che identifica se il cittadino eroga servizi alla comunità;
- contatori di ore settimanali (o altro intervallo temporale di riferimento impostato nel codice sorgente), che tengono conto delle ore effettivamente utilizzate da un cittadino per un dato servizio. Tali contatori vengono utilizzati come condizione perché un cittadino possa erogare/fruire il servizio, nelle funzioni '*trade_function*' e '*select_customer*';
- i livelli di conoscenza che i vari agenti hanno tra loro;
- i parametri del TPB.

Le funzioni nella classe dei cittadini sono:

- *relative_agreement_interactions*: tale funzione esegue le interazioni che avvengono tra agenti dentro e fuori la comunità secondo la dinamica di opinione del RA. Gli agenti che eseguono questa funzione sono i cittadini attivi all'interno della comunità. Questa funzione viene attivata ad ogni step dagli agenti interni la comunità; tuttavia, è possibile limitarne l'esecuzione ad un dato numero di step.
- *social_pressure_interactions*: simula le interazioni, intese come sola influenza sociale, come previsto dal TPB per quanto riguarda la pressione sociale (che influenza la *Subjective norm*). Ad ogni iterazione in cui la funzione viene richiamata, ciascun agente viene influenzato da un certo numero di altri agenti con opinioni forti appartenenti al proprio social network. Tale funzione viene eseguita all'interno della funzione '*TPB*' e, quindi, come descritto in seguito, verrà attivata solo da cittadini esterni alla comunità. Le opinioni forti rispettano le seguenti condizioni: livello di conoscenza con self maggiore di '*kn_thsld*', incertezza minore di 0,7, e valore assoluto del livello di opinione maggiore di

0,6. Tali valori possono essere modificati nel codice sorgente e devono essere calibrati su dati acquisiti da casi studio reali.

Una volta determinati gli agenti che esercitano un'opinione forte sull'agente in esame (*self*), vengono applicate le equazioni Eq.2 ed Eq.3 del TPB che esprimono la pressione sociale che un certo agente esercita su *self*, da cui viene calcolato con Eq.4 il valore medio della pressione sociale esercitata da tutti gli agenti da cui *self* è influenzato. La funzione restituisce il valore \underline{sp} se $N \geq 1$, altrimenti restituisce il valore nullo per indicare che *self* non è stato influenzato da alcun agente.

- *TPB*: esegue le equazioni descrittive della teoria TPB per simulare il processo psicologico e sociale dei cittadini propedeutico alla loro eventuale adesione alla comunità, determinando quanti cittadini aderirebbero rispetto ad una popolazione campione presa come riferimento. Nell'attuale versione del codice, la funzione viene eseguita solo per gli agenti esterni alla comunità ed è finalizzata a determinare il loro eventuale ingresso nella comunità. Nella funzione vengono richiamati tutti i termini, relativi all'agente che la esegue (*self*), che compongono le equazioni del TPB, siano essi statici (*inf*, *pb*) o dinamici (*att*, *sn*), sulla base delle equazioni descritte nella sezione precedente. Viene pertanto calcolato con Eq.8 il valore della *behavioral intention*.

Infine, viene calcolato il valore del comportamento finale con Eq. 9. Il valore del comportamento finale viene confrontato con il parametro del modello b_{thslid} per stabilire se l'agente *self* intenda entrare a far parte della comunità ($b > b_{thslid}$).

- *initialize_new_citizens*: utilizzata nella classe dei servizi, questa funzione serve per identificare come erogatori e fruitori, rispetto ai vari servizi, i cittadini che entrano a far parte della comunità durante la simulazione.
- *select_customer*: lo scopo di tale funzione, che viene passata come argomento della funzione '*trade_function*', è quello di restituire l'erogatore migliore per il servizio *i*-esimo per un dato agente fruitore in un dato step temporale. L'agente che attiva questa funzione deve essere identificato come fruitore per il servizio considerato, considerando il livello di conoscenza tra erogatore e fruitore.
- *trade_function*: funzione che esegue le transazioni tra gli agenti cittadini (nell'attuale versione cittadino-cittadino e cittadino-comunità). Tale funzione è pensata per essere generica in modo tale da rendere modulari le funzioni presenti nella classe dei servizi, che attivano '*trade_function*'. A valle dell'inizializzazione di alcune variabili (ammontare di token per la transazione, token fee escluse, numero di ore, contatori delta transazioni), la funzione verifica due condizioni: la prima viene rispettata se la transazione considerata è del tipo cittadino-comunità, mentre la seconda se la transazione è tra cittadini (allo stesso modo si può mettere un'ulteriore condizione quando verrà aggiunta la classe degli stakeholder, al fine di considerare anche gli scambi cittadino-stakeholder). In entrambi i casi la funzione aggiorna vari contatori al verificarsi di determinate condizioni simulando così il corretto avvenimento della transazione.

- o Cittadino-comunità: le condizioni che devono essere rispettate sono:

- il cittadino è erogatore di servizi verso la comunità;
- ore disponibili per servizi alla comunità;
- ore settimanali disponibili;
- ore richieste dalla comunità.

- o Cittadino-cittadino: le condizioni che devono essere rispettate sono:

- L'agente (cittadino) è fruitore del servizio;
- L'agente ha token sufficienti per fruire del servizio;
- Son rispettati i vincoli sulla disponibilità di ore;
- Deve essere disponibile un erogatore.

Se tutte le condizioni sono rispettate, il modello simula la transazione ed aggiorna i seguenti contatori: wallet del fruitore e dell'erogatore, contatore delle fee dell'erogatore, contatore delle transazioni e delle ore fruite/erogate dal fruitore/erogatore.

- *azzera_contatori*: tale funzione viene attivata se è rispettata la condizione per cui gli step della simulazione sono multipli del parametro temporale del modello (ad ora imposto essere uguale a 7, ovvero la settimana). Nel caso la condizione sia rispettata, dunque ad ogni intervallo temporale imposto, vengono resettati i pertinenti contatori di ore.
- *demurrage*: tale funzione serve per l'attivazione dell'omonimo meccanismo, che viene attivato nel caso in cui il wallet personale di un cittadino sia superiore all'attributo e parametro del modello '*demurrage_threshold*'. Nel caso in cui i token nel wallet del cittadino siano superiori alla soglia definita dall'utente, i token eccedenti alla soglia sono trasferiti automaticamente nel fondo di solidarietà della comunità.

2.1.6.4 Services

La classe dei servizi raggruppa tutti i servizi inclusi nella 'Smart Token Economy'. Nell'attuale versione del codice sono considerati tre servizi: i primi due rientrano nella categoria di servizi mutualistici tra cittadini, mentre il terzo raggruppa i servizi cittadino a comunità. Le modalità con cui è stata implementata tale classe, ed il modo in cui è stata collegata alla classe dei cittadini, rende le funzioni presenti in questa classe modulari e ripetibili. In questo modo si è cercato di semplificare l'implementazione futura di ulteriori servizi generici. Gli attributi nella classe dei servizi la collegano alle classi SmartTokenEconomy, Community e Citizens. Per ciascun servizio è implementata una funzione con la quale vengono interfacciati i cittadini tra loro e con la comunità, sono aggiornati i contatori delle transazioni e sono trasferite le informazioni che vengono utilizzate per il plottaggio dinamico dei risultati nella GUI MESA.

2.1.6.5 Smart token economy

Tra gli attributi ci sono tutti i parametri del modello sopra descritti. Oltre ad essi vi sono i seguenti:

- *schedule*: serve per collegare la classe del modello con lo scheduler;
- *grid*: per la visualizzazione di una griglia nell'interfaccia grafica di MESA;
- *active_citizens*: per tenere traccia del numero di cittadini attivi presenti nella comunità;
- *non_active_citizens*: per tenere traccia del numero di cittadini esterni alla comunità;
- *datacollector*: per il salvataggio dei dati (model-level e agent-level) sotto forma di dataframe, successivamente salvati ed elaborati sfruttando istanze già esistenti nella classe 'DataCollector' di MESA;
- *community*: collega la classe del modello con quella della comunità;
- *services*: collega la classe del modello con quella dei servizi;
- *running*: imposto come 'True', ossia la simulazione prosegue fino a quando l'utente non la stoppa tramite il relativo pulsante posto nell'interfaccia grafica, azione che aggiorna l'attributo in 'False'.

Le Funzioni presenti nella classe sono di seguito descritte:

- *minimum_initial_population*: impone un numero minimo di cittadini attivi all'istante iniziale (pari al valore minimo tra tre e la metà della popolazione iniziale) nel caso in cui non ci siano agenti interni alla comunità, il che si potrebbe verificare in caso di ridotta popolazione impostata nella GUI.
- *stop_running*: funzione usata per interrompere la simulazione nel caso in cui per 14 step consecutivi (equivalenti a due settimane di tempo reale) non avvenga nessuna transazione. Per ogni step in cui non avvengono transazioni l'attributo della comunità '*stop_running*' aumenta di 1, e così di seguito fino a che non raggiunge il valore 14, quando l'attributo del modello '*running*' diventerà 'False' e interromperà la simulazione. Se però avviene una transazione il contatore si resetta e riparte da 0.

2.1.7 Visualizzazione dei grafici e GUI MESA

La sezione della GUI di MESA che riporta i grafici è composta da una griglia che rappresenta schematicamente la popolazione degli agenti interni ed esterni alla comunità e da una serie di grafici che sono selezionati per

visualizzare dinamicamente l'evoluzione della comunità, in termini di adesioni, flussi di token ed euro, opinione degli agenti esterni alla comunità per una valutazione della tendenza a aderire alla stessa.

La griglia degli agenti (Figura 3) rappresenta il potenziale bacino di utenti della comunità, riportando con colori diversi gli agenti che fanno parte della comunità da quelli che sono esterni alla stessa ma che interagiscono con i membri della comunità e potrebbero aderire alla stessa. La posizione degli agenti nella griglia è puramente indicativa, ovvero non rappresenta la reale posizione degli stessi nella comunità.

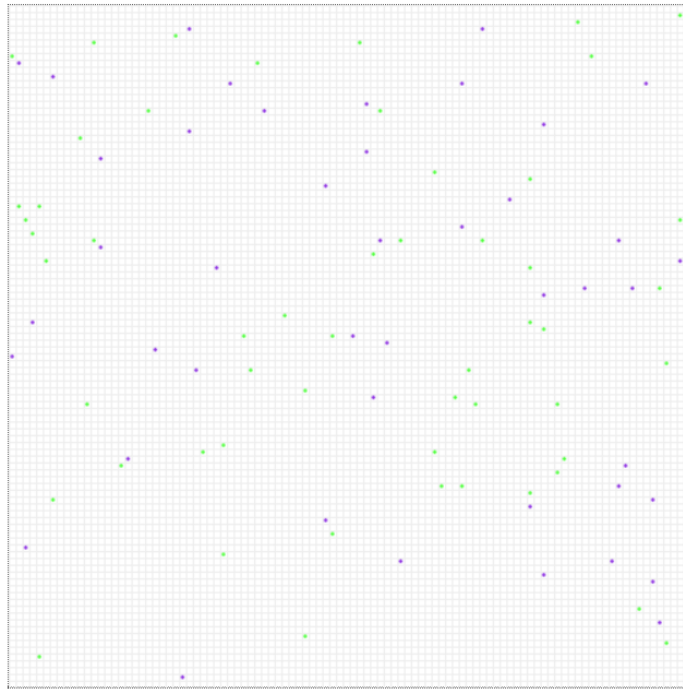


Figura 18: Griglia degli agenti della GUI di MESA

Segue il grafico normalizzato a 100 dei cittadini che aderiscono alla comunità (cittadini attivi) e di quelli che sono esterni alla comunità (cittadini non attivi), riportato in Figura 19. Il grafico include l'andamento dei cittadini non attivi con un'opinione $opi > 0,5$, rappresentativa del consolidamento di opinione da parte dei cittadini e quindi del numero residuo di agenti che potrebbe entrare nella comunità (man mano che diminuiscono gli agenti esterni minore è la probabilità che l'opinione di coloro che non hanno aderito alla comunità cambino idea).

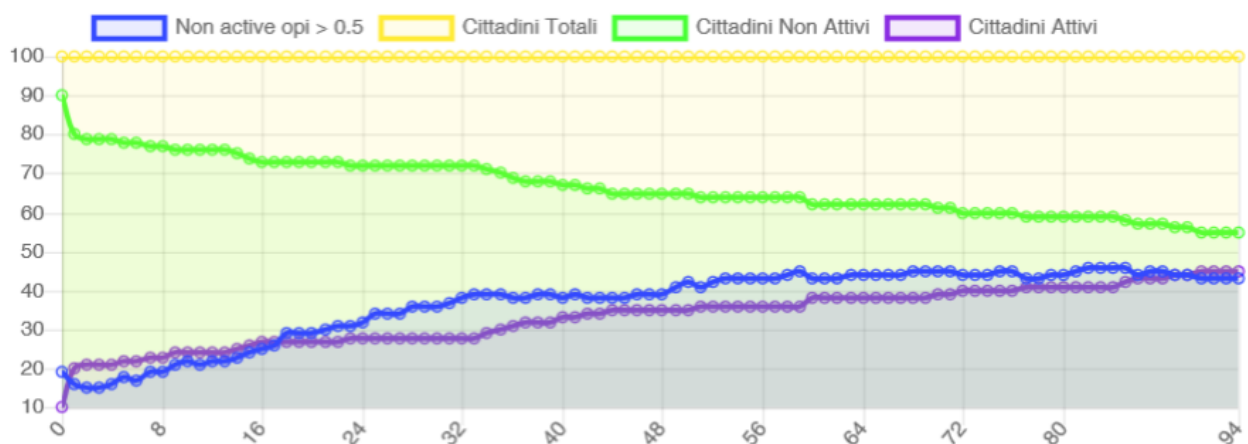


Figura 19: Andamento degli agenti in base alla loro adesione alla comunità

In Figura 20 è riportato l'andamento medio dell'opinione e della relativa incertezza per gli agenti esterni alla comunità in un dato intervallo di tempo. Nell'attuale versione del simulatore, l'opinione e l'incertezza relativa degli agenti che aderiscono alla comunità assumono valori tipici di opinioni forti e non vengono più modificati. In altri termini, il modello assume che la loro opinione non cambi più e che abbiano un elevato grado di soddisfazione. Per questo motivo l'andamento dell'opinione media dei cittadini attivi è costante e pari a 0 nel grafico sotto riportato (è posta pari a zero in quanto omessa). Un possibile sviluppo del simulatore potrebbe consistere nel modellare la dinamica di adeguamento dell'opinione anche per i membri interni alla comunità, ad esempio in base all'andamento delle loro transazioni e dello scambio dei beni/servizi.

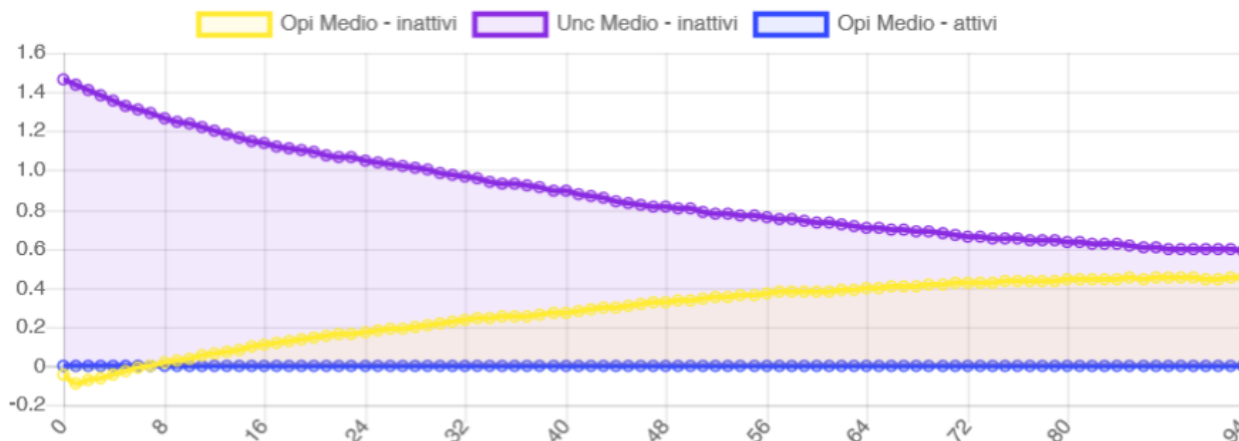


Figura 20: Andamento medio dell'opinione e della relativa incertezza degli agenti esterni alla comunità

In Figura 21 sono confrontati gli andamenti cumulati nel tempo dei token scambiati, delle transazioni e delle tasse nella comunità, mentre in Figura 22 è riportato l'andamento delle transazioni e dei token scambiati ad ogni intervallo di tempo. L'evoluzione nel tempo delle transazioni per i singoli servizi tra cittadini (nella versione attuale sono simulati due servizi tra i cittadini) e dei servizi che i cittadini erogano a favore della comunità è riportata in Figura 23, mentre il confronto tra gli ingressi (total fee, euro fund, fondo di solidarietà) e le uscite (total bonus adesione, servizi alla comunità), in termini sia di token che di valuta a corso legale (euro) per la comunità sono riportati in Figura 24.

Oltre ai grafici, i risultati – in termini di wallet e di transazioni per ciascun servizio per ogni singolo agente e per la comunità per ogni intervallo di tempo – sono salvati anche in file in formato .csv e .txt.

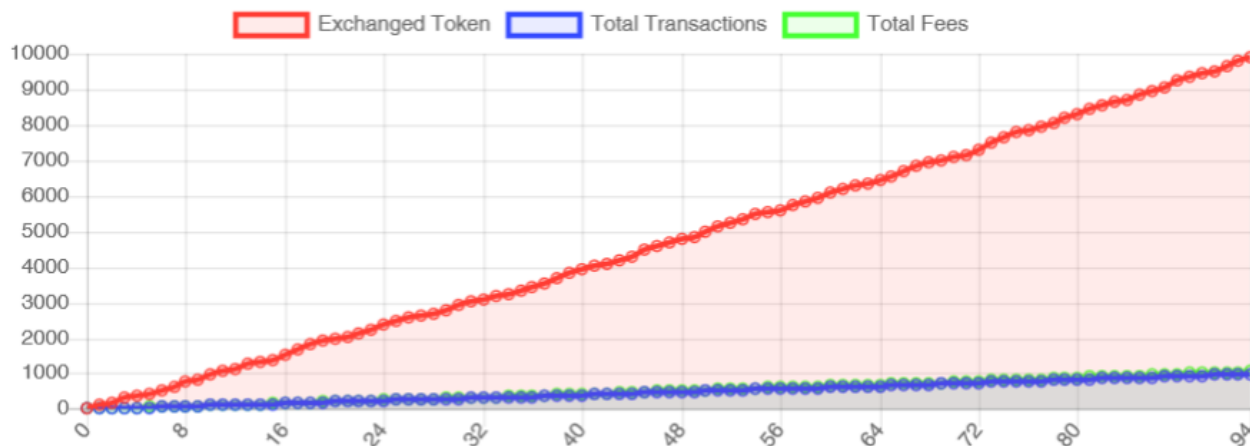


Figura 21: Confronto tra l'andamento cumulato dei token scambiati, delle transazioni e delle tasse nella comunità

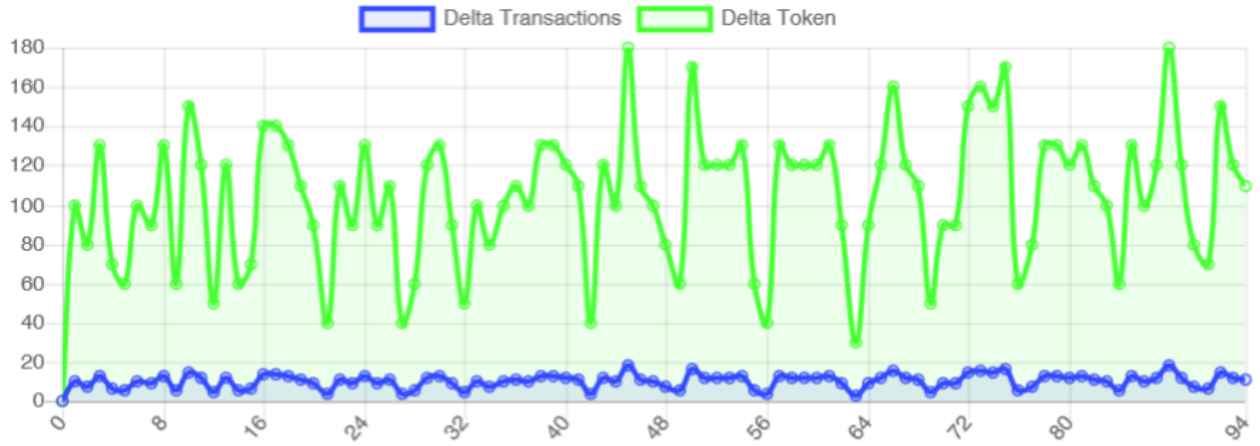


Figura 22: Andamento delle transazioni e dei token scambiati ad ogni intervallo di tempo

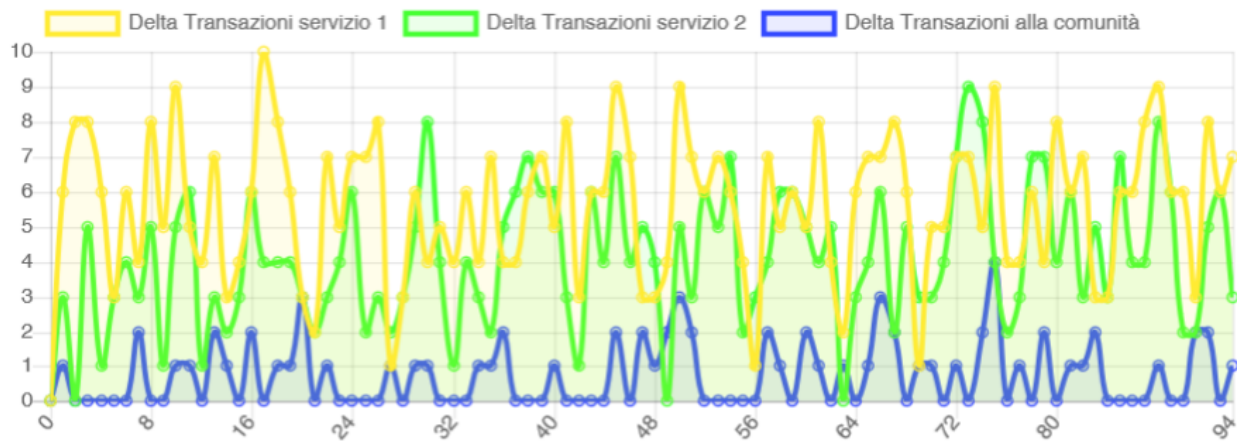


Figura 23: Andamento delle transazioni per i vari servizi attivi tra cittadini e verso la comunità

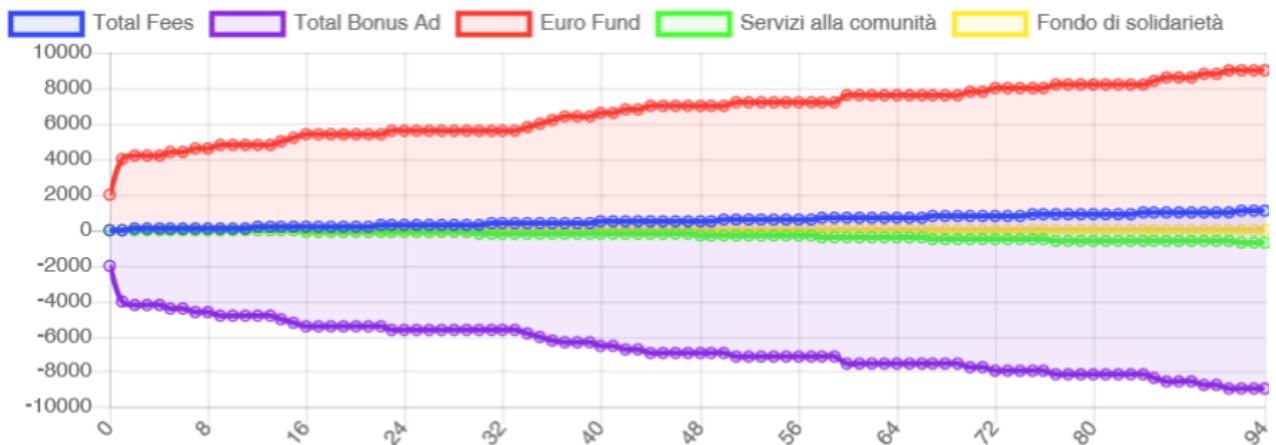


Figura 24: Andamento a livello della comunità degli ingressi (total fee, euro fund, fondo di solidarietà) e delle uscite (total bonus adesione, servizi alla comunità), in termini sia di token che di euro

2.1.8 Analisi di sensitività sui parametri del modello del Relative Agreement

È stata svolta un’analisi di sensitività per valutare come viene influenzato il comportamento del modello ad agenti al variare dei due parametri tipici del modello RA, che simula la dinamica dello scambio di opinioni [5]:

- ϕ corrisponde al numero di interazioni, ossia di opinioni scambiate, per ciascun agente che avvengono in uno step temporale;
- μ regola la velocità di convergenza delle opinioni.

L’analisi è stata svolta fissando uno dei due parametri e variando l’altro secondo intervalli predefiniti. Tale procedura è stata eseguita con il fine di valutare l’influenza di entrambi i parametri sulla dinamica di opinione che è stata implementata nel modello ad agenti.

Durante le simulazioni effettuate, gli altri parametri iniziali del modello sono stati mantenuti costanti per limitare l’analisi ai soli parametri caratteristici del RA. Tuttavia, a causa della natura stessa del modello ad agenti, molti fattori interni al modello potrebbero influenzare i risultati ottenuti. Tra questi fattori sono sicuramente presenti gli attributi randomici che definiscono lo stato interno di ciascun agente del modello, che possono essere legati in qualche modo al RA (livello di conoscenza, distribuzione delle opinioni e rispettive incertezze) ma non solo (numero di ore settimanali pro capite disponibili per servizio, attributo di confronto per l’esecuzione dello scambio dei servizi).

In Tabella 6: Parametri iniziali del modello per l’analisi di sensitività sono mostrati i parametri iniziali imposti al modello durante le simulazioni.

Tabella 6: Parametri iniziali del modello per l’analisi di sensitività

Popolazione totale [numero agenti]	100
Popolazione attiva a t=0 [numero agenti]	10
Fruitori Servizio 1 [%]	50
Fruitori Servizio 2 [%]	40
Erogatori alla comunità [%]	30
Bonus di adesione ai cittadini [token]	200
Totale ore disponibilità cittadini [h]	500
Disponibilità settimanale massima cittadini [h]	10
Disponibilità settimanale massima per comunità [h]	2
Max ore settimanali fruibili dalla comunità [h]	40
Tassazione [%]	13
Limite di demurrage [token]	1000

La variabile del modello che è stata scelta per valutare l’influenza dei parametri ϕ e μ è il numero di cittadini che aderiscono alla comunità. Tale scelta deriva dal fatto che il numero di cittadini che aderiscono alla comunità è influenzato direttamente dall’algoritmo che descrive la dinamica delle opinioni; in quanto l’adesione viene abilitata nel momento in cui la loro opinione, che dipende dal RA, supera un valore di soglia impostato.

In entrambe le analisi svolte, per ciascun set di valori (ϕ , μ) sono state effettuate tre diverse simulazioni per una durata di 130 step ciascuna; in seguito, è stato calcolato il valore medio sulle tre simulazioni del parametro di interesse.

2.1.8.1 Analisi di sensitività del parametro ϕ

Tale analisi è stata eseguita mantenendo costante la velocità di convergenza μ pari a 0,1. Il numero di interazioni ϕ è invece fatto variare con step di 2; sono considerati cinque valori di ϕ , rispettivamente: 2, 4, 6, 8, 10. In Figura 25 e in Figura 26 sono mostrati i risultati delle analisi effettuate. In generale tutte le curve mostrate sono caratterizzate da un andamento descrivibile in tre fasi: nella prima la crescita delle adesioni è limitata e graduale; la seconda è la fase transitoria in cui avvengono la maggior parte delle adesioni; mentre nella terza e ultima fase le curve si stabilizzano sul valore di saturazione, oltre al quale la comunità non cresce. È possibile notare come, con l'aumentare del numero di interazioni tra agenti ϕ , diminuisca la durata della prima fase ed aumenti la pendenza della curva nella seconda. Questo comportamento è particolarmente accentuato tra $\phi = 2$ e $\phi = 4$. Il numero massimo di cittadini che entrano a far parte della comunità calcolato dal modello varia leggermente negli scenari analizzati, tuttavia è plausibile che questo risultato dipenda da altri aspetti del modello ad agenti piuttosto che dal RA.

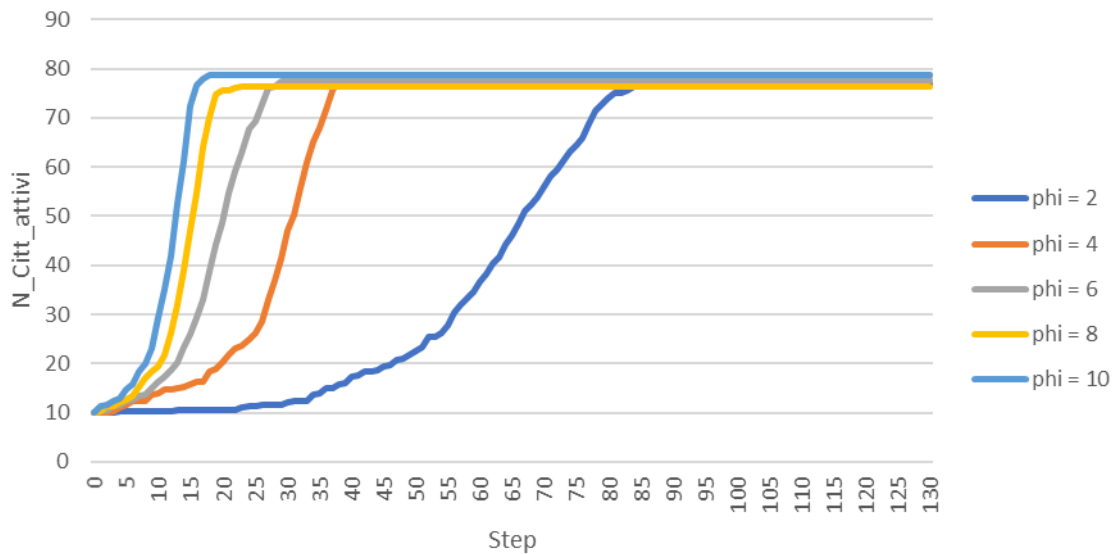


Figura 25: Sensitività del modello al parametro ϕ

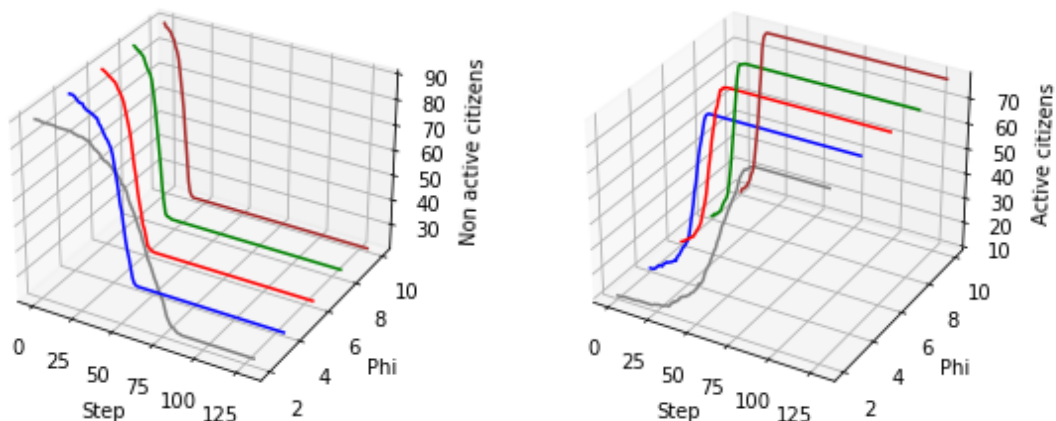


Figura 26: Andamento del parametro ϕ per i cittadini esterni (non attivi) e interni (attivi) alla comunità

2.1.8.2 Analisi di sensitività del parametro μ

In questo caso il parametro ϕ è mantenuto costante e pari a 2, mentre μ è fatto variare con uno step di 0,1 nell'intervallo compreso tra 0,1 e 0,5 (estremi inclusi). In Figura 27 sono mostrati i risultati ottenuti. Analogamente al parametro ϕ , le curve ottenute sono caratterizzate da tre distinte fasi, inoltre al crescere di μ si riduce la durata della prima fase e la curva diventa più pendente. Le differenze riscontrate tra le varie curve sono evidenti per valori di μ minori, mentre quando μ assume valori tra 0,3 e 0,5 il cambiamento è meno accentuato. In tutti gli scenari analizzati il valore di saturazione della popolazione che entra a far parte

della comunità è raggiunto in momenti diversi. In Figura 28 sono mostrati i risultati ottenuti in grafici tridimensionali dove il numero di cittadini che aderiscono alla comunità (cittadini attivi) è plottato in funzione dello step temporale e del parametro μ ; per completezza è plottato anche il complementare dei cittadini attivi, ovvero i cittadini che rimangono esterni alla comunità.

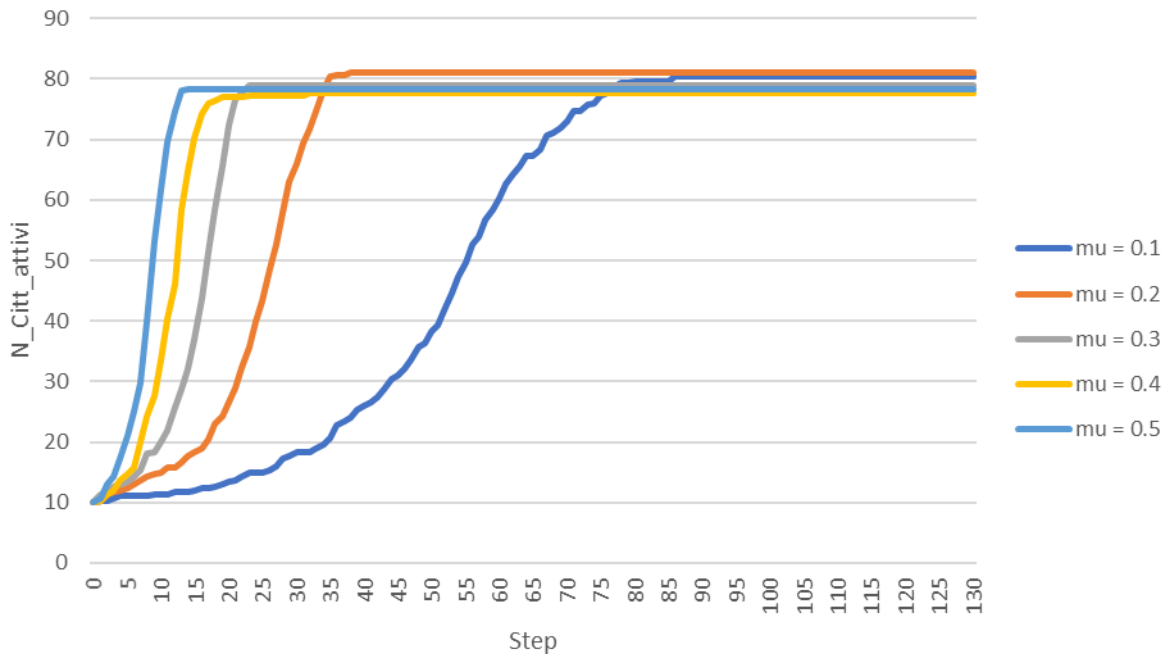


Figura 27: Sensibilità del modello al parametro μ

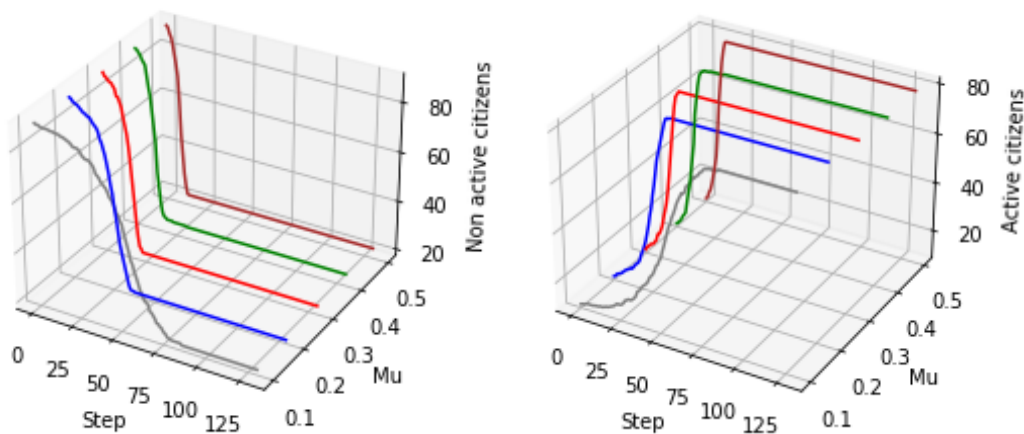


Figura 28: Andamento del parametro μ per i cittadini esterni (non attivi) e interni (attivi) alla comunità

3 Qualificazione marketplace

Come illustrato nel report RdS/PTR(2021)/039, l'implementazione iniziale del market place utilizza uno strumento commerciale sviluppato dalla ditta EXTRADE che lo fornisce come Software as a Service (SaaS) con una fee annuale. Questa scelta è stata dettata dalla necessità di dimostrare la fattibilità e l'usabilità di una soluzione software che permetta la realizzazione e la gestione di un marketplace all'interno di una Local Energy Community, piuttosto che lo sviluppo da zero di una simile soluzione. Questo ha permesso sia di ridurre di molto i costi della messa a disposizione di questa soluzione ma, soprattutto, di avere dei tempi compatibili con la realizzazione del progetto.

L'eventuale sviluppo economico di questa soluzione è, però, lasciato alla ditta così come la volontà di utilizzarlo all'interno di una comunità, sostenendo i relativi costi di gestione, rimane in carico alla Comunità stessa.

L'obiettivo di questo studio, quindi, non è dimostrare la funzionalità del software in quanto tale ma la sua applicabilità a scenari tipici che si possono realizzare nella gestione degli scambi di beni e servizi in una Local Energy Community.

Per semplificare l'adattamento e ridurre al massimo il rischio di inserire errori nella piattaforma, si è scelto di utilizzare la piattaforma blockchain già utilizzata dall'applicazione Trakti. In questo modo, è stato semplificato il processo di sviluppo e implementazione degli smart contract necessari per lo studio.

Lo scenario analizzato riguarda la generazione e lo scambio di token legato alla applicazione di un contratto già sottoscritto da un fornitore di servizio e da un utente. Tale contratto viene rispettato da entrambe le parti e, in base alle politiche adottate dalla Comunità e dagli accordi tra le parti contenute nel contratto, una certa quantità di token viene generata e un'altra scambiata tra gli utenti.

Per supportare questo scenario, è stato implementato su Trakti il contratto di distribuzione token e le dApps di controllo e monitoraggio del contratto e dei token assegnati secondo lo scenario definito, qui mostrate con dati di test. In Figura 29, sottostante, si vedono i token assegnati sulla base della campagna attivata e sottoscritta. Ogni singola transazione può essere esplorata al fine di vederne i dettagli.



Figura 29: Andamento dei token di un utente

Nello scenario attuale, l'utente che fornisce un servizio viene remunerato in token per la fornitura del servizio stesso e, in base al servizio proposto, può essere ulteriormente remunerato dalla Comunità, ad esempio nel caso di fornitura di servizi a scopo sociale o legati al riutilizzo di un bene che altrimenti sarebbe diventato un rifiuto. Dall'altra parte, l'utente che acquisisce un servizio lo paga in token ma può ricevere dalla Comunità

un rimborso, sempre in token, legato alla sostenibilità del suo acquisto. Ad esempio, nel caso in cui acquisti un bene riciclandolo o riutilizzandolo.

La piattaforma tiene traccia di due differenti quantitativi di token: quelli disponibili e quelli in attesa (pending): questi totali sono calcolati sulla base delle letture acquisite dalla blockchain e le somme pertanto spettanti, ma che devono essere ancora distribuite. Come si può notare potrebbero esserci una serie di campagne contratti sottoscritti in grado di generare una distribuzione di token nel tempo.



Figura 30: variazione dei token in base alla sottoscrizione dei contratti nel tempo

In Figura 30 viene mostrato l’andamento dei token legati alla realizzazione di uno o più contratti (pagamenti ricevuti) con altri utenti della comunità. Questi pagamenti sono diretti, stabiliti dal contratto e vengono effettuati nel momento in cui il pagamento viene autorizzato dall’acquirente.

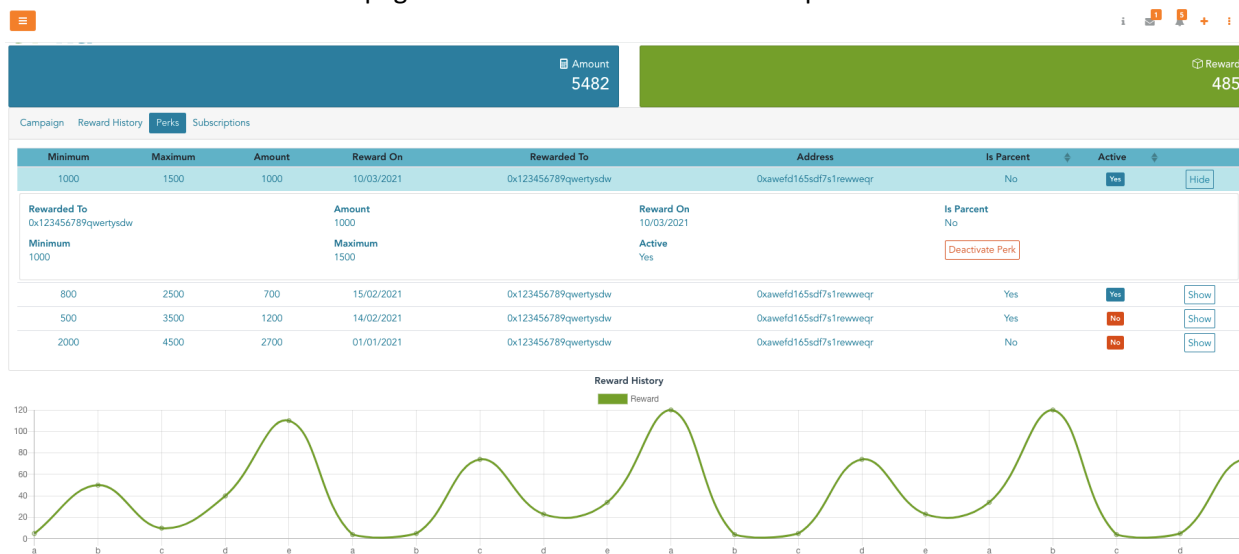


Figura 31: Dettaglio dei bonus ricevuti

La Figura 31 presenta invece l’andamento dei token ricevuti come “bonus” per i servizi offerti o i beni acquistati nella logica di supportare quelle azioni ambientalmente ed energeticamente vantaggiose per la

comunità o l'ambiente. Questi token vengono erogati dalla comunità e sono "creati" in base a delle regole interne e criteri stabiliti dal gestore: l'erogazione non è immediata ma viene calcolata periodicamente.

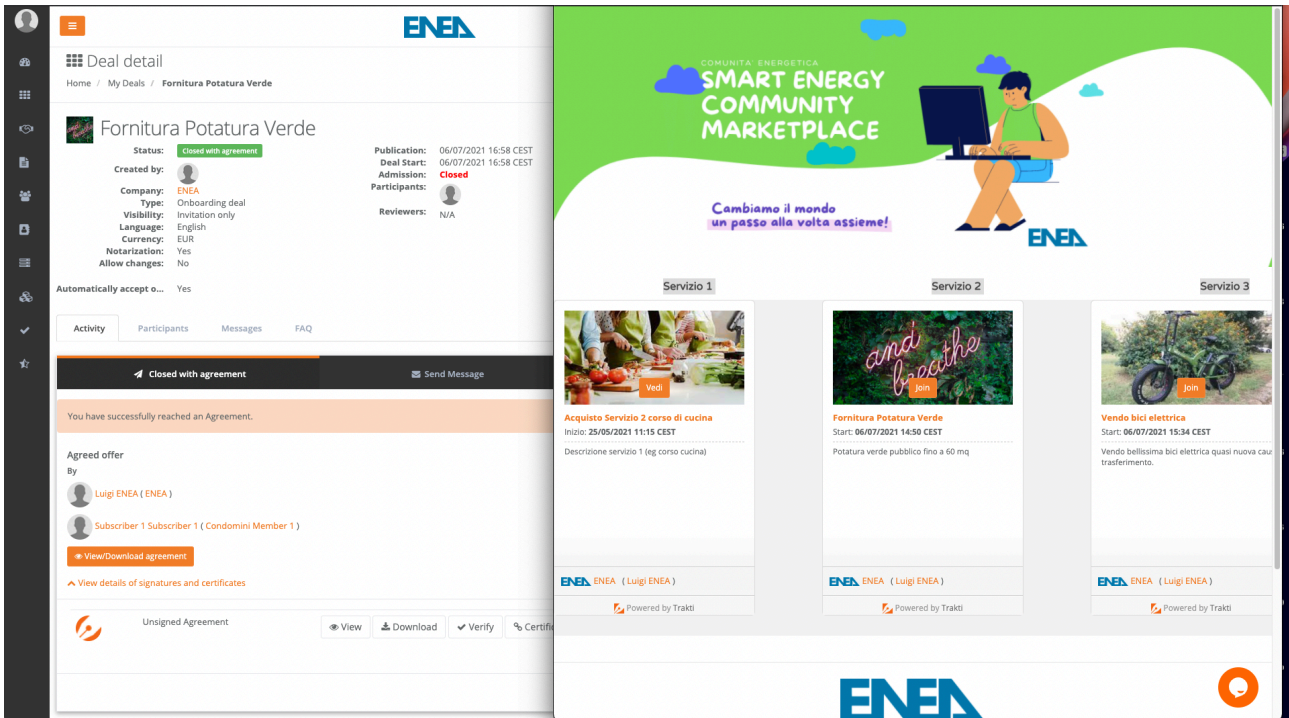


Figura 32: Pubblicazione offerta e dashboard di gestione e vista dell'ENEA Marketplace

La Figura 32 mostra una schermata di inserimento di un'offerta sul marketplace, in questo caso la "Potatura del Verde" da parte di un offerente e la schermata con le offerte disponibili che un acquirente può scegliere in base alle proprie necessità.

4 Qualificazione Eclistener

Come illustrato nel rapporto tecnico 1.48, ECListener si avvale di una molteplicità di piattaforme:

- il cluster Kubernetes in produzione, che chiameremo brevemente “Cluster”
- la piattaforma cloud di sviluppo e testing, costituita da un cluster OpenStack, che chiameremo “Arena”, per evocare il concetto di un insieme di “sand-box” cioè di piattaforme su cui sperimentare.
- il server standalone di sviluppo denominato “babylon” (essendo raggiungibile tramite l’indirizzo pubblico babylon.casaccia.enea.it)

Nel seguito tratteremo la qualificazione delle varie componenti hardware, software e di alcuni algoritmi utilizzati.

4.1 Qualificazione del Cluster

La prima qualificazione riguarda le capacità prestazionali; in questo ambito, diverse sono le caratteristiche principali su cui ci focalizzeremo:

- le capacità di immagazzinamento dati (storage);
- la potenza computazionale;
- il backup e ridondanze raid
- la continuità di servizio che conferisce la resilienza al sistema.

Storage

come detto le tre macchine che costituiscono il cluster sono un server e due workstation. Tutte sono accessibili tramite diversi indirizzi internet corrispondenti a diverse reti per diverse esigenze. In particolare tutte e tre le macchine fanno parte della rete interna ENEA del centro Casaccia denominata *intranet.casaccia* ed accessibile solo dall’interno o tramite VPN.

Il server (denominato Ur e quindi accessibile da *ur.intranet.casaccia*) ospita i seguenti dischi:

- un’unità a stato solido SSD (Solid State Drive) da 900GB che ospita esclusivamente il sistema operativo e il software di base.
- due unità SATA da 2TB in modalità Raid1, equivalenti quindi ad un disco da 2TB totalmente ridondato dal controller (Dell Perc). Questa unità ospita il filesystem /home in cui risiedono i dati degli utenti che in tal modo sono sempre ridondati e quindi la loro accessibilità è resiliente rispetto alla perdita di un disco per danni aleatori, ma non ovviamente per manipolazioni deliberate (come ad esempio attacchi ransom-ware).
- Una unità SATA da 2TB allocata alla memoria condivisa sul cluster descritta in seguito.

Ognuno dei due server (denominati Nippur e Kish, raggiungibile rispettivamente da *nippur.intranet.casaccia*, *kish.intranet.casaccia*) ospita i seguenti dischi:

- due unità SATA da 2TB in modalità Raid1, equivalenti quindi ad un disco da 2TB totalmente ridondato dal controller. Anche in questo caso, come per il server, questa unità ospita il filesystem /home in cui risiedono i dati degli utenti che sono quindi ridondati.
- Una unità SATA da 2TB allocata alla memoria condivisa sul cluster descritta in seguito.

Tutti i dischi fisici possiedono una velocità di accesso pari a 7200RPM (giri al minuto) compatibile con i controller raid installati.

Tutti i dischi sono sostituibili in modalità “hot plug” e possono quindi essere sostituiti senza spegnere le macchine o interrompere i servizi.

Il Cluster è dotato di una unità di immagazzinamento distribuita tra le tre macchine che lo compongono, realizzata tramite il software Ceph (<https://docs.ceph.com/en/pacific/rados/index.html>). Ogni macchina rende disponibile un disco fisico da 2TB per costituire un’unica unità condivisa accessibile da tutti con

indirizzamento locale in configurazione distribuita “RAID5” di rete: ogni dato è replicato in almeno due supporti fisici in modo tale che la perdita o manomissione di uno qualsiasi di essi non comprometta la disponibilità dei dati. Chiameremo questa risorsa “unità ceph”. Ovviamente la ridondanza imposta limita la capienza totale dell’unità a 3TB.

Nella configurazione iniziale (quella prevista dalla progettazione) l’unità ceph ospita le banche dati su cui si basano i macro-servizi; in tal modo la disponibilità dei dati è “robusta” rispetto alla compromissione di una qualsiasi macchina o disco allocato all’unità ceph.

Anche dal punto di vista della potenza computazionale occorre distinguere le diverse prestazioni delle due tipologie di componenti:

- il server possiede memoria ad accesso random da 64GB (4 bank da 16 GB RDIMM, 2933MT/s, Dual Rank) ed un processore Xeon Gold da 3GHz e 48 core.
- ognuna delle workstation possiede memoria ad accesso random da 64GB (4 bank da 16 GB RDIMM, 2933MT/s, Dual Rank) ed un processore Xeon Silver da 3GHz e 24 core.

L’uso dell’unità distribuita ceph rappresenta lo standard per l’architettura a docker su kubernetes adottata, che è stata utilizzata inizialmente proficuamente.

Durante la fase di produzione (ovvero acquisizione e trattamento dei dati) si sono verificati frequenti blackout dovuti ad ampliamenti della rete del centro programmati e altri accidentali; tali eventi sono stati utilizzati per testare le componenti. La sperimentazione eseguita suggerisce una configurazione di un cluster CEPH con l’aggiunta di almeno due nodi (per un totale di 5 nodi) e la configurazione di software di backup utilizzati in ambienti Kubernetes come, ad esempio, Velero (<https://velero.io/>). La configurazione attuale, utilizzata per lo sviluppo delle attività progettuali, può essere considerata una configurazione minimale. Come già sottolineato in precedenza la creazione di un cluster in grado di sopravvivere anche in presenza di malfunzionamenti severi (‘failure’) che coinvolgono più nodi, necessiterà ulteriori miglioramenti:

- l’allocazione di almeno due nodi master duplicati (mirror)
- la distribuzione geografica dei cluster duplicati (su macchine fisicamente distanti)
- software di allineamento e backup come il già citato Velero.

L’esperienza sui blackout accidentali ha consigliato una configurazione delle banche dati più conservativa e si è preferito trasformare la configurazione da k8 (kubernetes) alla più tradizionale Postgres. La banca dati postgres è anche raggiungibile tramite un secondo indirizzo sulla rete pubblica della ricerca GARR2 corrispondente al nome jerico.casaccia.enea.it, quindi sul dominio enea.casaccia (il cui anonymous system è gestito dall’ENEA).

Backup

I backup pianificati per il cluster sono di due tipi: un backup incrementale giornaliero (automatizzato) ed alcuni “dump integrali” non automatizzati gestiti in base alle esigenze ed in particolare a seguito del raggiungimento di milestones come, ad esempio, il passaggio alle banche dati postgres. Il backup incrementale è realizzato tramite degli script di *cron*, mentre i dump sono realizzati con il comando di linea “pgdump”.

Il backup incrementale è attualmente limitato ai codici su una macchina geograficamente distinta (halab.dpcpnet.casaccia) collocata sulla rete interna casaccia dhcpnet. I codici utilizzati per i dati twitter sono invece modificati raramente e ridonati solo nei dump.

È importante sottolineare che il backup giornaliero dei dati acquisiti non è particolarmente utile in quanto i prodotti utilizzati consentono l’ispezione retrograda nel tempo fino a sei mesi: un tempo maggiore dei dump integrali.

La banca dati di Mongo DB era inizialmente basata su Ceph e quindi automaticamente ridondata, nella fase attuale è necessario predisporre un ulteriore backup autonomo che tuttavia non è particolarmente urgente essendo i documenti utili pertinenti memorizzati nella banca dati postgres.

Monitoraggio del cluster

Il cluster Kubernetes è stato dotato di una serie di strumenti per il controllo delle prestazioni e il monitoraggio del funzionamento dello stesso. Il primo strumento aggiunto al cluster è gestito tramite la classica dashboard di Kubernetes che consente di visualizzare lo stato di funzionamento del cluster e monitorare lo stato di funzionamento di tutte le risorse gestite dal cluster stesso.

La dashboard di Kubernetes è raggiungibile all'indirizzo <https://jerico.casaccia.enea.it:31366> (e quindi anche dalla retepubblica GARR2) e l'autenticazione è basata su token come mostrato nella seguente figura Figura 33.

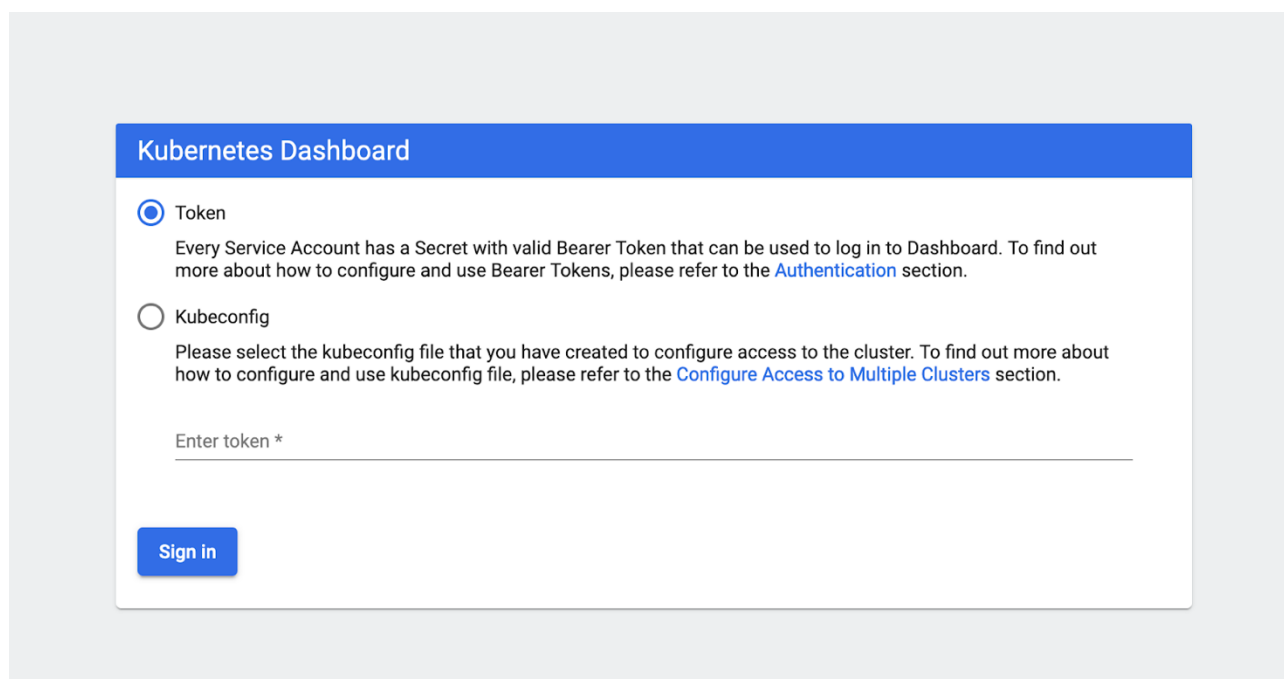


Figura 33: Login della dashboard di Kubernetes

La Figura 34 mostra le risorse nell'istante dell'ispezione "consumate" dal deployment del servizio EListener.

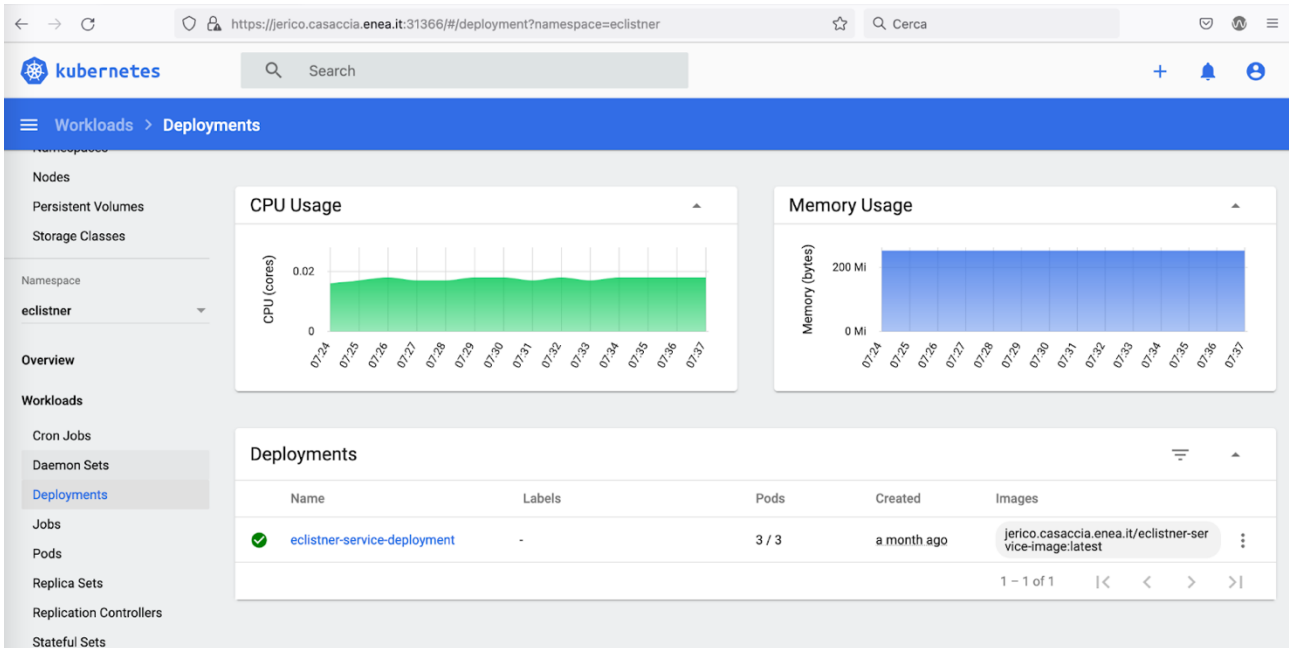


Figura 34: Dashboard Kubernetes. Panoramica generale delle risorse consumate dal cluster

La Figura 35 mostra le risorse consumate dai tre pods generati dalle repliche previste dal deployment di EListener.

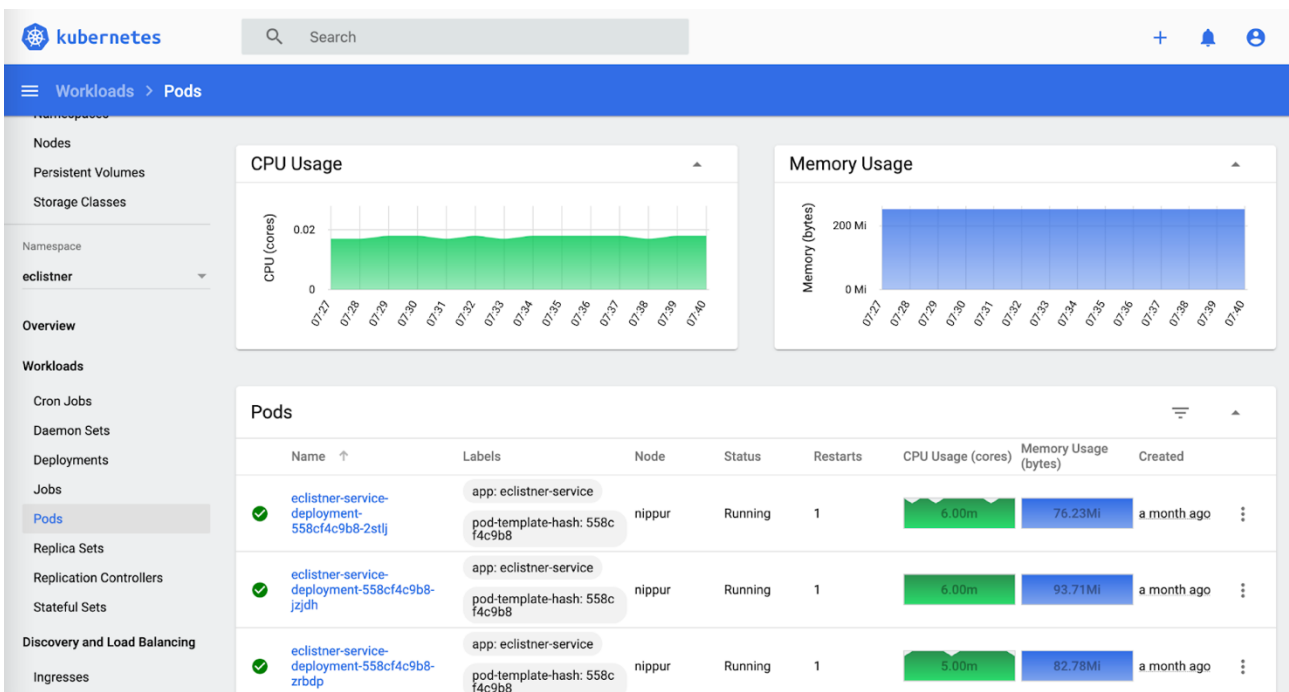


Figura 35: Dashboard Kubernetes. Risorse consumate dai pods dell'applicazione MediaBot

Il cluster è dotato anche della risorsa aggiuntiva denominata “metric-server” (<https://github.com/kubernetes-sigs/metrics-server>) che consente un monitoraggio a linea di comando utilizzando il comando esteso `kubectl top` non disponibile nell’installazione minimale di un cluster kubernetes. La Figura ECL4 mostra l’output del comando `kubectl top` per la visualizzazione delle risorse consumate dai vari “pods” del namespace `twitterms`. E’ utile notare che kubernetes introduce un ulteriore livello di “namespace” che consente di ospitare i microservizi che compongono l’applicazione TwitterBot allocando ad ogni pod un ip di una rete virtuale 10.100.0.0/16.

E’ anche importante notare che il cluster è dotato di una rete interna fisica reale 10.0.0.0/32 che consente le comunicazioni tra le macchine fisiche anche in difetto della rete interna intranet.casaccia (192.168.152.0/24). Si tratta, in questo caso, di una resilienza rispetto al malfunzionamento della rete interna.

```
(base) [root@ur ~]# kubectl top pods -n twitterms
NAME                                CPU(cores)   MEMORY(bytes)
consumer-deployment-6cf5c5c96d-tbt7d 7m           16Mi
extract-deployment-dc87c7fb8-z4n4m    4m           607Mi
filtered-deployment-6cdc955788-czkdf  4m           11967Mi
keywords-deployment-6ddf6f8748-ssdbt  5m           73Mi
twitter-test-687f677949-f6vqt        9m           30Mi
```

Figura 36: Esempio output del comando `kubectl top` abilitato da `metric-server`

Come è possibile vedere dalla Figura 36, Kubernetes definisce le risorse consumate (e allocate ai diversi pods) in termini di “unità CPU”, per quanto riguarda le risorse di calcolo, e bytes per le risorse di memoria. Una unità CPU è equivalente ad un core fisico, nel caso di installazione bare metal, o virtuale, nel caso di installazione su macchine virtuali. È possibile richiedere (e conseguentemente mostrare sulla dashboard) frazioni di unità di CPU. Ad esempio il pod denominato “consumer” (descritto in seguito) consuma “7m CPU (cores)”, come mostrato nella Figura ECL4. Questo significa che richiede l’allocazione di 0.007 CPU poiché 1m CPU equivale a 0.001 CPU. Quindi, nel momento fotografato nell’immagine, il servizio non sta consumando risorse significative del cluster. La situazione cambia al ricevere di un flusso importante di tweets come è mostrato nel rapporto tecnico 1.50 (*Test e sperimentazione della infrastruttura e dei servizi per le LEC*).

L’utilizzo della risorsa “metric-server” è fondamentale per l’abilitazione di altre applicazioni di monitoraggio come Prometheus e Grafana.

Prometheus è installato come servizio sul nodo Nippur raggiungibile all’indirizzo <http://192.168.154.95:32201> (nippur.intranet.casaccia). Con questo tool, che rappresenta lo standard de facto per il monitoring di un cluster Kubernetes, è possibile visualizzare e analizzare diverse metriche come ad esempio:

- metriche classiche di sysadmin riguardanti i nodi del cluster: cpu, disco, memoria;
- metriche a livello di orchestrazione come lo stato dei deployment, le risorse richieste, la latenza delle API di Kubernetes etc.;
- Metriche interne delle componenti di kube-system (componenti base di Kubernetes) come, ad esempio, il DNS server (<https://sysdig.com/blog/kubernetes-monitoring-prometheus/>).

Grafana invece è solo una interfaccia user-friendly per Prometheus.

Prometheus e Grafana, insieme con il tool di visualizzazione di dashboards sono stati installati ed è prevista una loro “customizzazione” per visualizzare ed analizzare le performance del cluster Kafka utilizzato dal workflow TwitterBot e da possibili workflow futuri. Questa customizzazione è in fase di studio nell’ambiente di sviluppo Arena (il cloud open-stack).

La Figura 37 mostra un esempio di visualizzazione di due grafici in Prometheus relativi al cluster in produzione. In alto si possono visualizzare i pods attualmente in stato di esecuzione (running) per i diversi nodi, mentre in basso il grafico mostra il numero di processi “container” attivi nei tre nodi Ur, Kish e Nippur.

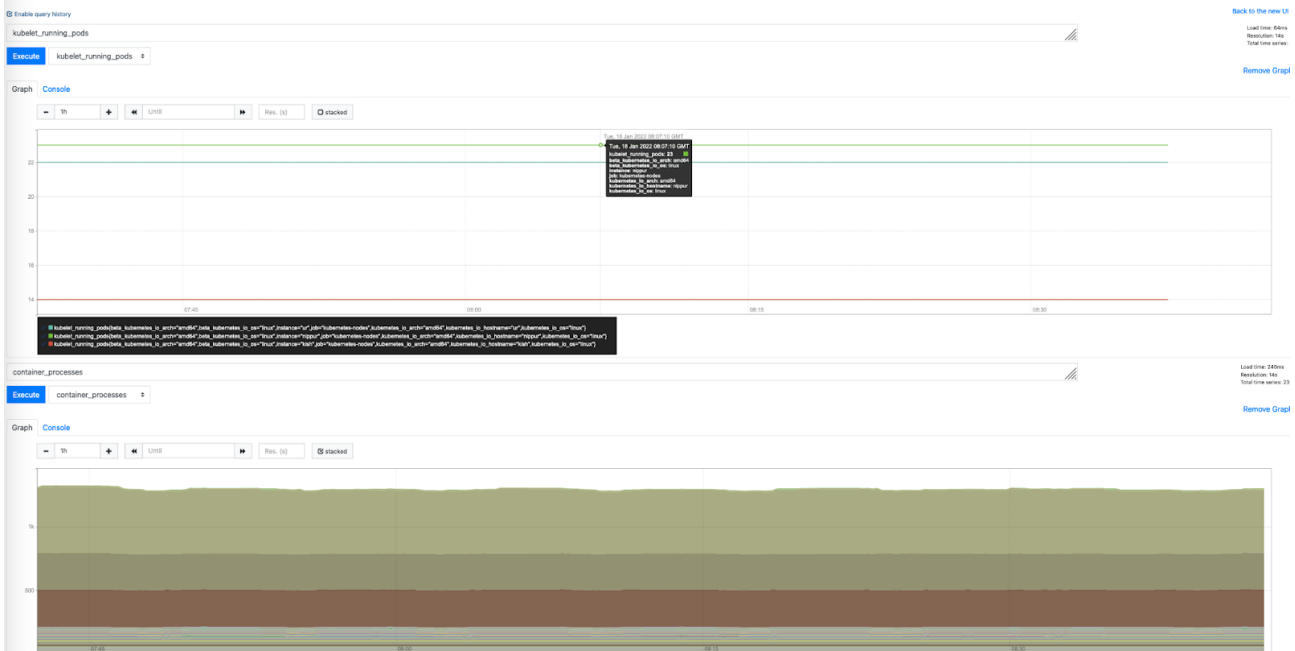


Figura 37: Dashboard Prometheus

Resilienza della piattaforma informatica

Ricordando i già citati limiti fisici dell'hardware (configurazione minimale e assenza di duplicazione del nodo master), la piattaforma assicura un livello di resilienza soddisfacente per gli scopi attuali di progettazione e "produzione". Tuttavia, è auspicabile l'ampliamento della piattaforma stessa al fine di poter fornire servizi "cloud-based", con determinate caratteristiche di resilienza, per lo svolgimento delle varie attività progettuali future sulla Ricerca del Sistema Elettrico. Ad esempio, nel corso delle attività sono stati sperimentati alcuni operatori Kubernetes in grado di configurare e gestire macro-servizi Kubernetes per la configurazione, creazione e gestione di cluster PostgreSQL (Cruch Data Operator) e MariaDB (MariaDB Operator). Questi due operatori sono stati utilizzati con successo nei vari workflow operativi utilizzati. Il loro utilizzo comporta un'estrema facilità di configurazione e creazione del cluster desiderato in maniera analoga a quanto avviene oggi con l'utilizzo delle piattaforme cloud (a pagamento) più conosciute. I limiti attuali del cluster Kubernetes hanno suggerito l'utilizzo di servizi di storage esterni al cluster stesso per assicurare la persistenza e l'utilizzo dei dati per le analisi previste dal progetto. Infatti, l'accesso ai file system di Ceph, in fase di recovery dopo un evento indesiderato, anche se possibile, risulta essere molto complicato e quindi poco gestibile da un non esperto.

Dependability dell'applicazione MediaBot

L'applicazione MediaBot attualmente si basa su due microservizi gestiti da Kubernetes. Il primo servizio, denominato eclistener-collector, viene eseguito giornalmente e scarica il report di Google Alerts riguardante le web news pertinenti il dominio delle comunità energetiche selezionate dal servizio di Google secondo gli algoritmi proprietari molto efficienti, ma non resi noti. Il microservizio eclistener-collector è implementato nel namespace di default di kubernetes ed è gestito come un "CronJob" Kubernetes. In maniera simile al cronjob dei sistemi operativi Linux, il CronJob di Kubernetes esegue ad orari prefissati dall'utente un determinato script ovvero una applicazione. Kubernetes, all'orario specificato, crea un pod in uno dei nodi del cluster per eseguire il codice (nel nostro caso Python) dell'applicazione. La differenza sostanziale con il cronjob classico del sistema operativo, quindi, è che il codice in questo è eseguibile in uno dei nodi disponibili e non dipende da un solo nodo. In genere questo pod è eseguito sul nodo Kish; in ogni caso se questo non fosse disponibile, Kubernetes assegnerebbe il pod ad uno dei nodi operativi (Ur o Nippur). Quindi il processo è gestito a livello kubernetes e non di singola macchina.

Il secondo micro-servizio che analizza la webnews e la memorizza nel database PostgreSQL è stato realizzato come un web service REST (Representational state transfer). Tramite il framework FLASK (il più comune web-framework di python), sono stati creati degli endpoint URL (Universal Resource Locator) che possono essere invocati dalle altre applicazioni. Il servizio eclistner-collector, quindi, una volta recuperato il report di Google Alerts, chiama l'endpoint del servizio ECListener in modalità POST utilizzando come payload della richiesta il report di Google Alert stesso. Il servizio ECListener, sempre attivo ed in attesa di richieste, una volta ricevuta una richiesta POST (quindi nel protocollo http o https) analizza il report ed applica tutta la logica di business prevista dall'applicazione ECListener. Operativamente, come mostrato in Figura 38, il servizio ECListener è gestito da Kubernetes tramite un deployment con tre repliche.

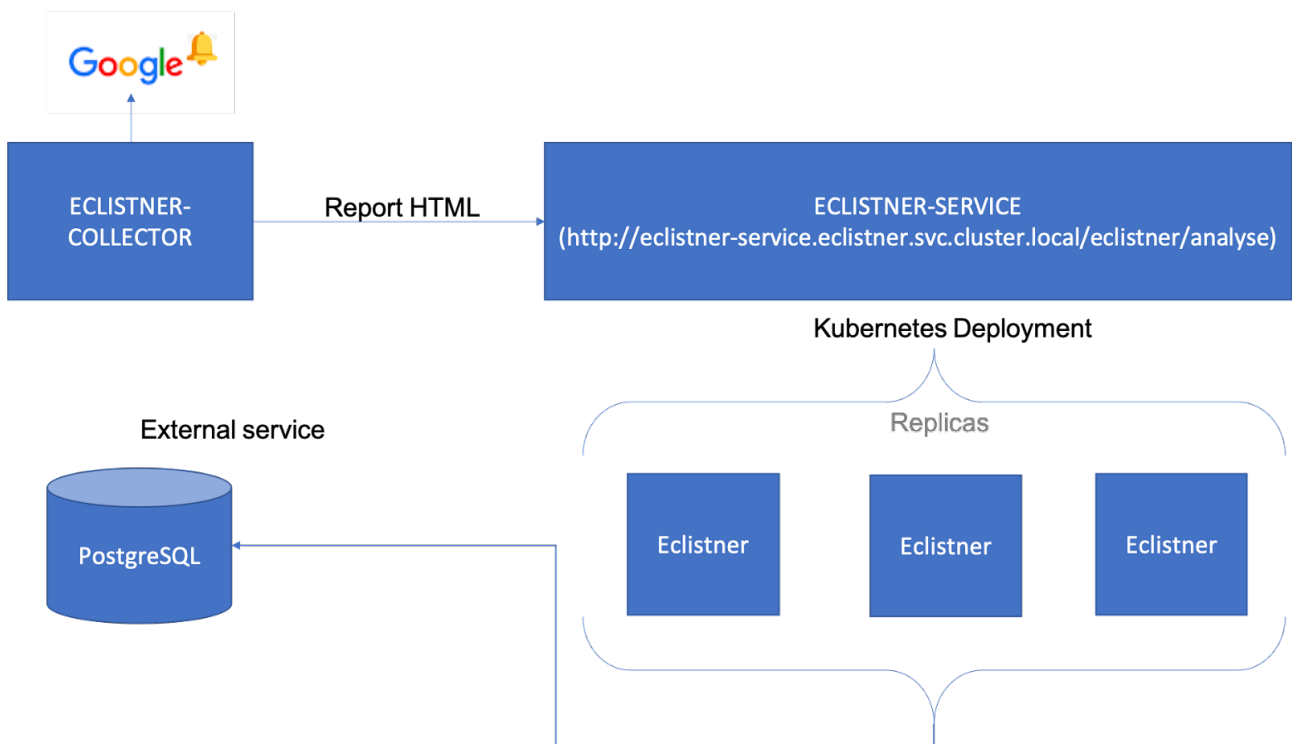


Figura 38: Architettura applicazione MediaBot

Questo assicura la continuità del servizio anche in presenza di un failure di uno o due nodi del cluster. Per questo workflow non è stata prevista l'introduzione del autoscaling automatico del numero di repliche del deployment. Tale meccanismo, introdotto nel workflow TwitterBot, consente a Kubernetes di gestire in maniera automatica lo scaling del numero di repliche nel caso in cui le repliche minime previste dal deployment non fossero sufficienti a gestire il carico di lavoro. Il flusso dati previsto dall'applicazione MediaBot, allo stadio attuale dello sviluppo del sistema, non è tale da richiedere questa modalità operativa.

Dependability applicazione TwitterBot

L'applicazione TwitterBot è stata ingegnerizzata seguendo il pattern architetturale a microservizi nell'ambito del contratto "Architetture distribuite, interoperabili ed affidabili per la realizzazione della piattaforma Obserbot nell'ambito della LEC" LA 1.65 e ampiamente descritto nel relativo rapporto tecnico. Nel documento corrente si riportano, brevemente, le caratteristiche di dependability del software realizzato. Analogamente a quanto mostrato precedentemente, per l'applicazione WebmediBot, tutti i microservizi che nel loro insieme contribuiscono alla realizzazione dell'applicazione TwitterBot sono stati implementati, come web service REST, utilizzando il framework FLASK di Python. Anche in questo caso, sono stati creati degli

endpoint URL che sono invocati, seguendo un determinato ordine, dagli altri servizi. I singoli microservizi possono essere anche chiamati individualmente per realizzare compiti specifici come quello di aggiornare l'insieme delle keyword. Per quanto riguarda l'applicazione TwitterBot la gestione dello streaming dei dati raccolti da Twitter è gestita tramite Apache Kafka. Anche in questo caso valgono le stesse considerazioni svolte per l'applicazione MediaBot riguardanti la gestione dello storage. L'applicazione TwitterBot utilizza i gestori di banche dati MongoDB e MariaDB per la memorizzazione di tutti i tweets: raccolti nel primo caso e i filtrati nel secondo. Nel corso delle attività progettuali sono stati sperimentati operatori per la gestione di questi DBMS (Data Base Management Systems) in maniera automatica nel cluster Kubernetes. I problemi riscontrati con il cluster CEPH, a seguito di vari blackout elettrici sperimentati dal server e l'impossibilità di disporre di un cluster configurato per avere la massima dependability, hanno comportato l'adozione di una soluzione esterna al cluster Kubernetes. Ad esempio, per quanto riguarda MongoDB è stata configurata una "Replica Set (RS)" utilizzando i tre nodi Ur, Kish e Nippur. Una replica set di mongod è costituita da un gruppo di processi "mongod" (il demone primario del sistema MongoDB) che condividono lo stesso insieme di dati. Una replica set ben configurata garantisce ridondanza dei dati (redundancy) e alta disponibilità (high availability). Il cluster MongoDB configurato tollera la perdita di uno dei server disponibile senza effetti considerevoli sulla disponibilità dei dati e del database.

La Figura 39 mostra la configurazione dell'applicazione TwitterBot gestita in maniera ibrida tramite servizi forniti dal cluster Kubernetes e dai servizi esterni di memorizzazione. Il dato non sperimentale, ma fornito dalla documentazione di MongoDB indica un intervallo di indisponibilità del replica set di MongoDB per la configurazione data pari a 10-12 secondi, in caso di perdita del nodo primario. Questo è, infatti, il tempo necessario dalla replica set per eleggere un nuovo primario ed abilitare le operazioni di scrittura; mentre le operazioni di lettura sono sempre disponibili anche se può verificarsi un temporaneo disallineamento con i dati inizialmente memorizzati sul primario. Per quanto riguarda il database MySQL, si sta indagando la possibilità di creare, anche in questo caso, un cluster con un numero fissato di repliche. Nel caso di MySQL, il requisito di disponibilità del servizio è meno stringente in quanto è possibile eseguire il lavoro di "filtering" (vaglio di pertinenza) in modalità differita (ed anche in batch) in caso di failure del DBMS. Rimane comunque essenziale predisporre un'attività periodica di backup dei dati memorizzati su questo database.

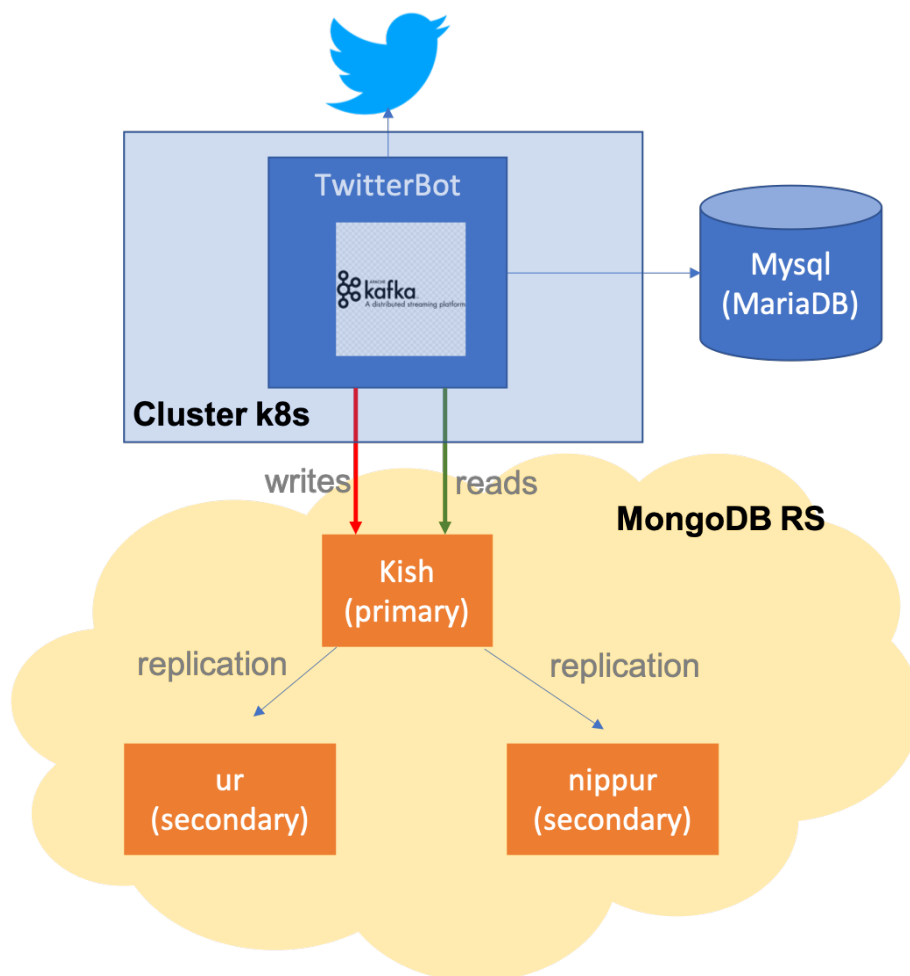


Figura 39: Architettura in produzione dell'applicazione TwitterBot

Resilienza rispetto alla perdita di alimentazione elettrica

Il Cluster è dotato di un UPS (Uninterruptible Power System, Dell Smart-UPS SRT 3000VA RM - UPS - 2700-watt - 3000 VA) della potenza di 3KVA in grado di sostenere l'intera piattaforma (incluso lo switch di rete digitale) per circa 10 minuti. Durante questo intervallo di tempo l'operatore umano viene raggiunto da un messaggio elettronico che allerta sul problema. Quando il livello di carica dell'UPS scende sotto un valore predefinito (in questo caso fissato al 3 minuti di autonomia) il software "power-chute" installato come server sull'UPS e come client su tutte le macchine fisiche (Ur, Kish e Nippur) provvede ad uno spegnimento dolce dei sistemi.

Piattaforme di sviluppo

Tutte le applicazioni sviluppate sono state prima realizzate e testate sotto altre due piattaforme allocate a questa funzione. Le due piattaforme sono chiamate rispettivamente "Arena" (un cloud privato Open stack) e Babylon (una macchina raggiungibile anche dall'esterno nella rete GARR2 allocata all'ENEA (cioè dominio casaccia.enea.it)).

La piattaforma Arena è costituita da 6 macchine HP Z2 ognuna basata su un processore intel I7 con 64GB di RAM, 2 dischi da 2TB 7200 rpm ed un disco da 4TB da 7200 rimovibile (per i backup).

Le 6 macchine formano un cluster "Open Stack" con una architettura tradizionale linux "open source" sviluppata dalla "Red Hat" e descritta nel sito da essi gestito (<https://www.redhat.com/it/topics/openstack>). Il sistema consente di creare macchine virtuali e reti virtuali per la loro connessione. Il cluster Kubernetes in produzione è stato clonato sulla piattaforma Arena consentendo lo sviluppo delle architetture a microsistemi

anche durante il periodo della pandemia che ha relegato le operazioni alla modalità remota. La Figura 40 mostra l'“Avatar” del cluster kubernetes sulla piattaforma Arena.

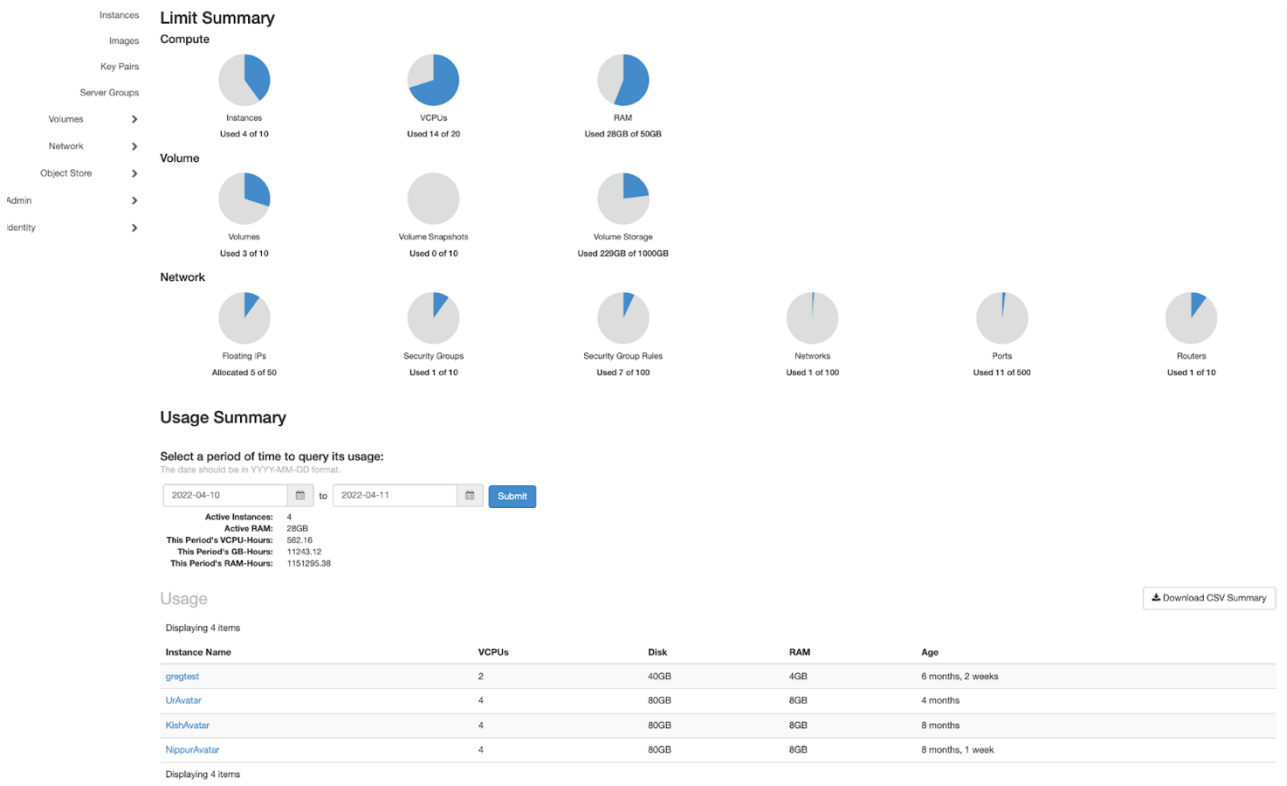


Figura 40: “Avatar” del cluster kubernetes sulla piattaforma cloud “open-stack” “Arena”.

La seconda piattaforma di sviluppo utilizzata è il server Babylon, raggiungibile sulla rete pubblica nel dominio casaccia.enea.it. Si tratta di un server Dell T640 dotato di un processore XEON gold da 64 core con 64GB di Ram e tre dischi da 7200rpm due in modalità raid 1 (controller perc) ed uno come unità di backup.

Il sistema operativo “Centos Stream 8” e tutto il software sono gli stessi delle macchine in produzione e tutto il software è sviluppato sotto docker per garantirne la facile portabilità. Anche questa macchina è dotata di un suo UPS e relativo software power-chute.

L’insieme di tutte le risorse allocate è un lavoro in itinere a supporto delle esigenze dei gestori delle local Energy Communities e dei ricercatori che ne studiano le proprietà generali, i quali hanno accesso ai web server.

5 Qualificazione dei modelli per l’elaborazione dei dati

Questa sezione verifica il funzionamento delle procedure e per il calcolo delle differenti grandezze che devono essere monitorate per il funzionamento delle Comunità energetiche e fornisce, utilizzando i dati raccolti dai sensori, i risultati grafici delle elaborazioni mostrando il codice operativo utilizzato per l’estrazione dei dati dal DB relazionale.

La verifica della correttezza delle informazioni e della validità dei dati è necessaria, prima di pubblicare queste informazioni attraverso il cruscotto CRUISE, per evitare la comunicazione di informazioni errate ai membri della comunità ed ai gestori della stessa.

Le query testate verranno poi rese disponibili, sotto forma di funzioni parametriche e complete di metadati descrittivi, al cruscotto CruISE per la pubblicazione verso gli utenti.

Per omogeneità e chiarezza, in tutte le query e le descrizioni è stata adottata la nomenclatura proposta in tabella seguente.

Tabella 7: definizione grandezze

Grandezza	Descrizione	Valori ammessi
T	Numero di valori quortorari in un giorno	96
$W_{i,g,t}$	carico quortorario (in Watt) dell'utente i , al giorno g , al tempo t (con t che varia da 1 a 96).	
$E_{i,g,t}$	consumo quortorario (in Wh) dell'utente i , al giorno g , al tempo t (con t che varia da 1 a 96).	
$N =$	numero di utenti nella comunità.	
D_{ins}	numero di dispositivi installati per la rilevazione di carico, corrente, ecc.	
M_{ins}	numero di dispositivi installati per la rilevazione meteo (temperatura, ecc.)	

5.1 Consumo giornaliero

Per il consumo giornaliero (kWh/giorno), si veda un esempio di grafico in Figura 41.

Il consumo giornaliero rappresenta il valore totale di consumo di energia per un singolo utente, in questo caso identificato solo con il numero interno del suo device, per ragioni di privacy. Tale consumo è confrontato con la media degli altri appartamenti di cui si raccolgono i dati. Come specificato, in questo caso il calcolo è svolto soltanto se il numero di dati raccolti in un certo giorno è maggiore del 90% dei dati giornalieri; quindi, se sono stati raccolti più di 86 dati in quel giorno. Questo permette di avere un valore ragionevole e non viziato da situazioni critiche. Nel grafico, i valori a 0 indicano i giorni in cui i dati raccolti non sono sufficienti ad operare il calcolo.

Consumo giornaliero dello specifico deviceid=5246

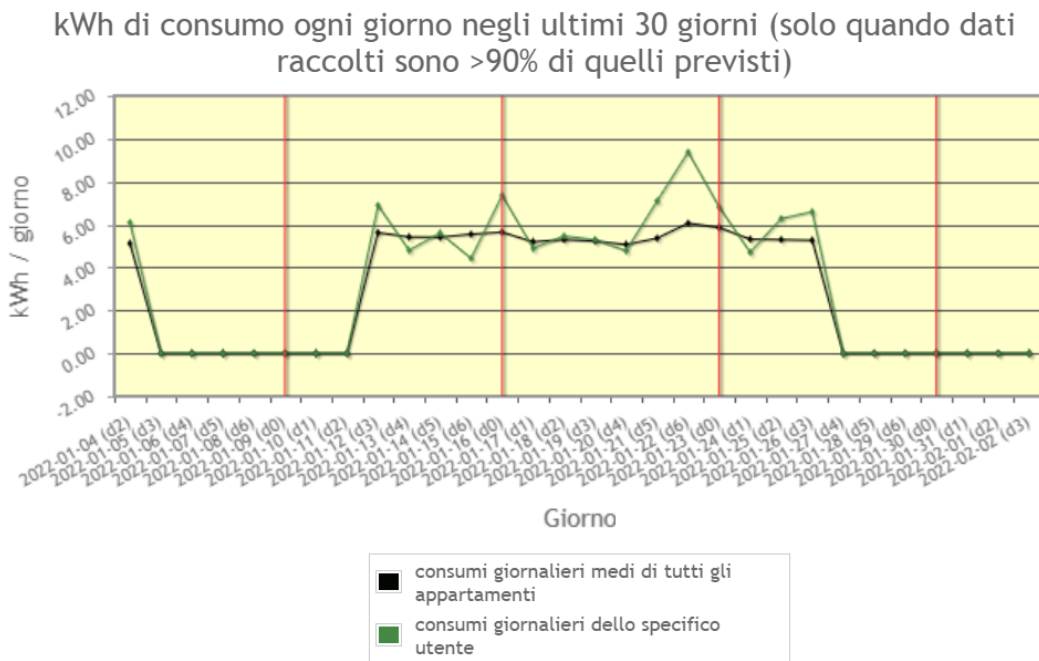


Figura 41: Esempio di consumo giornaliero

Il grafico in Figura 41 è il risultato di due operazioni distinte sui dati del database: il calcolo dei consumi giornalieri del singolo utente e il calcolo del valor medio per l'intera comunità.

1. Consumi giornalieri dello specifico utente

Equazione: consumo dell'utente i nel giorno g :

$$C_{ig}[\text{kWh/giorno}] = \frac{1}{1000} \sum_{t=1}^{96} E_{i,g,t}$$

Query:

```
SELECT *
FROM
  (SELECT ctid, CONCAT(timecomplete2, '(d',dayofweek,')') as timecomplete3, kwh3
  FROM vw_consumption_daily2
  where deviceid='%%P1%%'
  %%LIMIT%%) as orderlist
ORDER BY orderlist."timecomplete3" ASC
```

Parametri:

- %%P1%% indica il deviceid specifico dell'utente fissato .
- %%LIMIT%% è un 'parametro utente', cioè che, in futuro, potrà essere impostato dinamicamente dall'utente per estendere o ridurre il periodo di interesse. Nel grafico precedente, è posto uguale a 30.

2. Consumi giornalieri medi di tutti gli utenti della comunità

Equazione: consumo medio, nel giorno g , di tutti gli utenti della comunità

$$C_{gm}[\text{kWh/giorno}] = \frac{1}{1000} \sum_{i=1}^N \sum_{t=1}^{96} \frac{E_{ig,t}}{N}$$

Accesso: utente

Query:

```
Select * from (
  SELECT ctid, CONCAT(timecomplete, '(d',dayofweek,')') as timecomplete3, avgdailycons
  FROM vw_consumption_daily_avg_all2
  %%LIMIT%%) as orderlist
ORDER BY orderlist."timecomplete3" ASC
```

Parametri:

- %%LIMIT%% è un 'parametro utente', cioè che, in futuro, potrà essere impostato dinamicamente dall'utente per estendere o ridurre il periodo di interesse. Nel grafico precedente, è posto uguale a 30. Tale parametro deve essere coerente con quello scelto per la operazione precedente.

PROFILO DI CARICO MEDIO

Per il profilo di carico medio, si veda un esempio di grafico in Figura 42. Questo dato restituisce il valore medio della potenza richiesta dal singolo utente in una certa ora del giorno e la confronta con la media della comunità. In questo modo, è possibile tenere sotto controllo le richieste che la Comunità fa verso la rete elettrica nazionale e permette di capire la dimensione energetica della comunità e la sua domanda nel tempo. Fornisce informazioni anche sulle abitudini dell'utente.

Profilo di carico nell'ultimo anno

Comparazione tra media della comunità e consumo dell'utente singolo 55 nell'ultimo anno

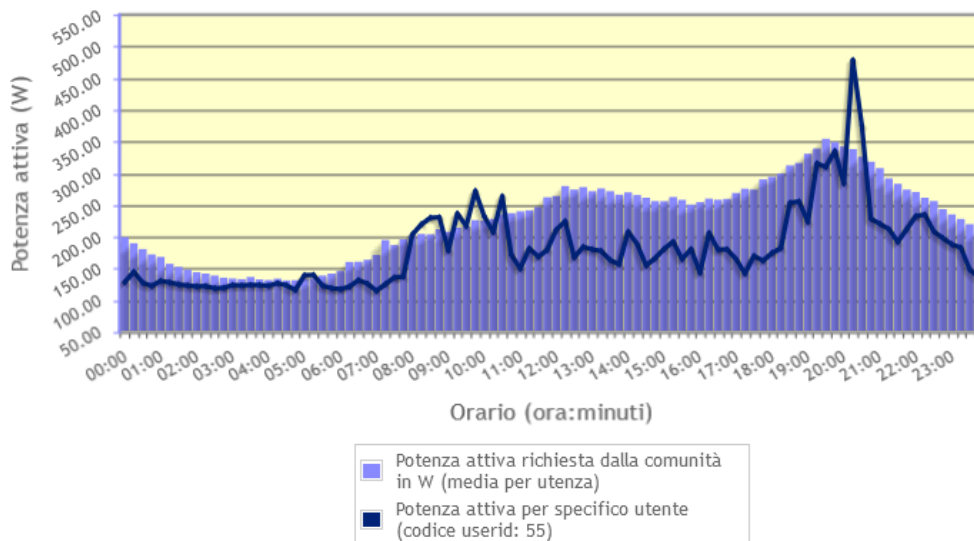


Figura 42: Esempio di profilo di carico medio

Per la creazione del grafico in Figura 42 sono necessari due insiemi di dati: la potenza attiva media richiesta dalla comunità misurata in Watt e la potenza attiva media per uno specifico utente.

1. Potenza attiva media richiesta dalla comunità misurata in Watt

Equazione: potenza attiva media, tra i giorni g_1 e g_2 , tra tutti gli N utenti, all'ora t :

$$P_{mt}[W] = \sum_{i=1}^N \sum_{g=g_1}^{g_2} \frac{W_{i,g,t}}{N(g_2-g_1)}$$

Query:

```
SELECT
    avg(powactive) as powactiveuseravg,
    time as orario
FROM public.vw_devices
LEFT OUTER JOIN vw_userconsumption on vw_userconsumption.deviceid=vw_devices.deviceid
    where usertype='Alloggio'
    and vw_userconsumption.communityid=%%communityID%%
    and vw_userconsumption.flowid=%%flowid%%
    and (timestamp >= CURRENT_TIMESTAMP - interval %%interval%%)
group by time
```

Parametri:

- %%communityID%% identifica la Comunità su cui si sta eseguendo l'analisi. Chiaramente, deve essere una di quelle in cui l'utente è attivo con un suo dispositivo
- %%flowid%% - identifica il flusso di informazioni quindi, di fatto, un device che viene monitorato
- %%interval%%) identifica il periodo in cui si vuole effettuare la valutazione.

2. Potenza attiva media per uno specifico utente

Equazione: potenza attiva, mediata tra i giorni g_1 e g_2 , dell'utente i , al tempo t :

$$P_{mt,i}[W] = \sum_{g=g_1}^{g_2} \frac{W_{i,g,t}}{(g_2-g_1)}$$

Query:

```
SELECT avg(powactive) as powactiveuser,
       time as orario
FROM public.vw_userconsumption
where communityid=%%communityid%% and userid=%%P2%% and flowid=%%flowid%%
and (timestamp >= CURRENT_TIMESTAMP - interval %%interval%%)
group by time
```

Parametri:

- %%cummunityID%% identifica la Comunità su cui si sta eseguendo l'analisi. Chiaramente, deve essere una di quelle in cui l'utente è attivo con un suo dispositivo
- %%flowid%% - identifica il flusso di informazioni quindi, di fatto, un device che viene monitorato
- %%interval%%) identifica il periodo in cui si vuole effettuare la valutazione
- %%P2%% identifica l'utente che viene monitorato.

Profilo di carico istantaneo della comunità

Per la potenza attiva, misurata in Watt, di tutta la Comunità permette di capire quanta potenza è richiesta dalla Comunità nel tempo. Questa potenza è necessaria, ad esempio, per capire se la produzione da fotovoltaico potrebbe rispondere a questa esigenza e quale potrebbe essere una dimensione corretta per le sorgenti. Un esempio è rappresentato nel grafico in Figura 43.

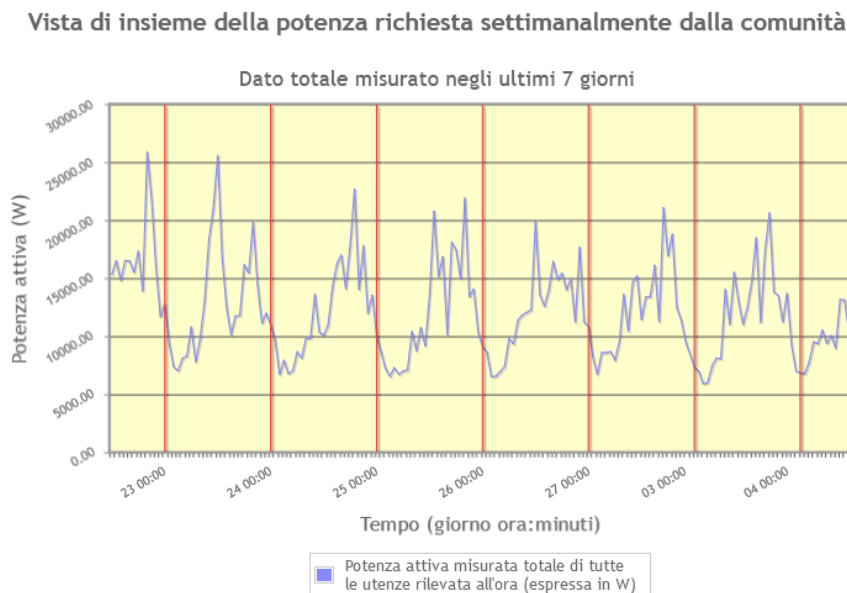


Figura 43: Esempio di profilo di carico istantaneo

Equazione: potenza totale della comunità nel giorno g , all'ora t :

$$P_{g,t}[W] = \sum_{i=1}^N W_{i,g,t}$$

Accesso: utente

Query:

```
SELECT *
FROM "vw_communitypowactivelast7days" ORDER BY "timestamp" ASC LIMIT 168
```

where communityid=%%communityid%%

Paramteri:

- %%communityid%%" identifica la comunità che si sta considerando per l'operazione

5.2 Anomalie – monitoraggio dei dispositivi non attivi

Per il monitoraggio dei dispositivi attivi, si veda un esempio di grafico in Figura 44.

Questa informazione permette di capire quanto dispositivi stanno fornendo informazioni rispetto a quelli installati e individuare eventuali anomalie nel sistema di raccolta delle informazioni.

Ad esempio, se sono installati 4 dispositivi, ma solo 3 sono attivi, il valore delle y plottato è di ¾=75%. Nell'esempio i dispositivi sono divisi in due classi, quelli che forniscono dati sulla produzione o il consumo elettrico e quelli che danno informazioni sul meteo locale. Queste informazioni sono disponibili solo per l'amministratore della piattaforma.

Il grafico in Figura 44 mostra i risultati congiunti di due operazioni, il calcolo del numero di dispositivi per il monitoraggio dell'energia elettrica e il numero di centraline meteo.

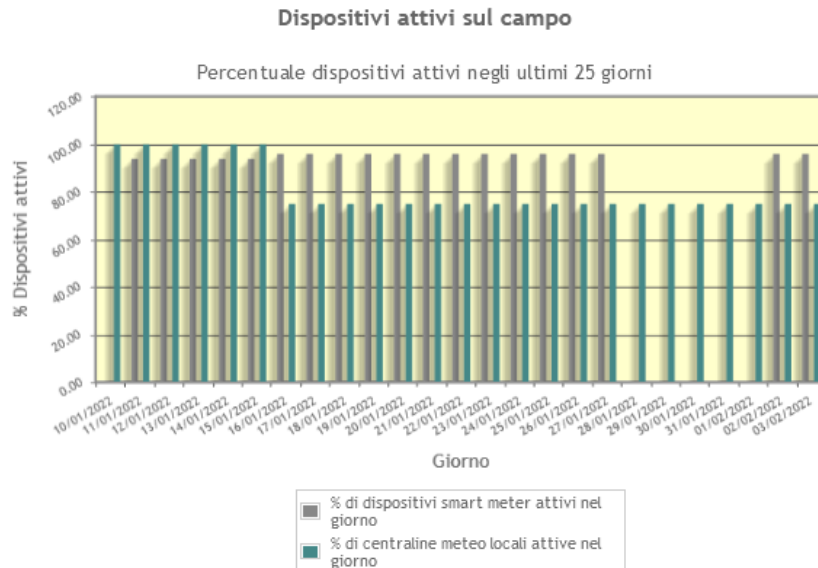


Figura 44: Esempio di monitoraggio di dispositivi attivi

1. Numero di dispositivi energetici attivi

Equazione: Dispositivi attivi

$$D_{ag}(\%) = 100(D_{fg}/D_{ins}),$$

dove D_{fg} sono i dispositivi funzionanti (con almeno un dato inviato in quel giorno) al giorno g , e D_{ins} i dispositivi installati.

Accesso: amministratore

Query:

```
SELECT *
FROM (select
    vws_scand_targetfrequency96normalised.*,
    coalesce(100*(vws_scand_targetfrequency96normalised.dailyactivedevices)
    vws_scand_targetfrequency96normalised.numdevicestotali,0) as percactivedevice
    FROM "vws_scand_targetfrequency96normalised" LIMIT %%NumberofDays%%) as orderedlist
ORDER BY orderedlist."timecomplete" ASC
```

Parametri:

- **%%NumberofDays%%** identifica il numero di giorni su cui si vuole fare il monitoraggio. Nell'esempio della figura, si è scelto di monitorare gli ultimi 25 giorni.

2. Frazione di centraline meteo attive nel giorno

Equazione: Dispositivi attivi:

$$M_a(\%) = 100(M_{fg}/M_{ins}),$$

dove M_{fg} sono i dispositivi meteo funzionanti al giorno g , e M_{ins} i dispositivi meteo installati.

Accesso: amministratore

Query:

```
SELECT *
FROM (
    select vws_scand_targetfrequencemeteo96normalised.*,
    100*(vws_scand_targetfrequencemeteo96normalised.dailyactivedevices)/vws_scand_targetfrequencemeteo96normalised.numdevicestotali as percactivedevice
    FROM vws_scand_targetfrequencemeteo96normalised
LIMIT %%NumberofDays%%
) as orderedlist
ORDER BY orderedlist."timecomplete" ASC
```

Parametri:

- **%%NumberofDays%%** identifica il numero di giorni su cui si vuole fare il monitoraggio.

5.3 Anomalie/funzionamento – monitoraggio dei dispositivi funzionanti

Il monitoraggio del comportamento dei dispositivi funzionanti permette di capire se il numero di messaggi inviati e ricevuti è quello atteso, ad esempio 96 messaggi al giorno per ogni dispositivo, oppure se ci sono dei malfunzionamenti per cui il numero dei messaggi è maggiore o minore di quelli attesi.

Esempio di anomalie di invio rispetto ai dispositivi attivi e funzionanti (quelli installati ma non funzionanti non sono contati a priori), mostrato in Figura 45. La figura mostra, per ogni giorno, la percentuale di messaggi ricevuti dai dispositivi attivi a funzionanti rispetto a quella attesa del 100%.

Ad esempio, se ci sono 4 dispositivi installati, ma 3 attivi e inviano 96 dati al giorno, la percentuale delle letture inviate è uguale al 100%.

Equazione: Funzionamento Dispositivi

$$AD(\%) = 100 \frac{V_{tg}}{D_{ins} \cdot 96}$$

dove V_{tg} è il numero di messaggi quattorari ricevuti ed immagazzinati con successo nel DataBase al giorno g , e D_{ins} i dispositivi installati. In assenza di anomalie, sono previsti 96 valori inviati al giorno per ogni dispositivo, e quindi $V_{tg} = D_{ins} \cdot 96$ valori. Nel caso in cui il dispositivo invii un numero differente di messaggi, il calcolo verrà parametrizzato correttamente rispetto al numero di invii attesi.

Anche in questo caso, la figura mostra sia la percentuale di messaggi provenienti dai dispositivi energetici che quella proveniente dalle centraline meteo. Anche in questo caso, occorrono 2 operazioni per generare il grafico.

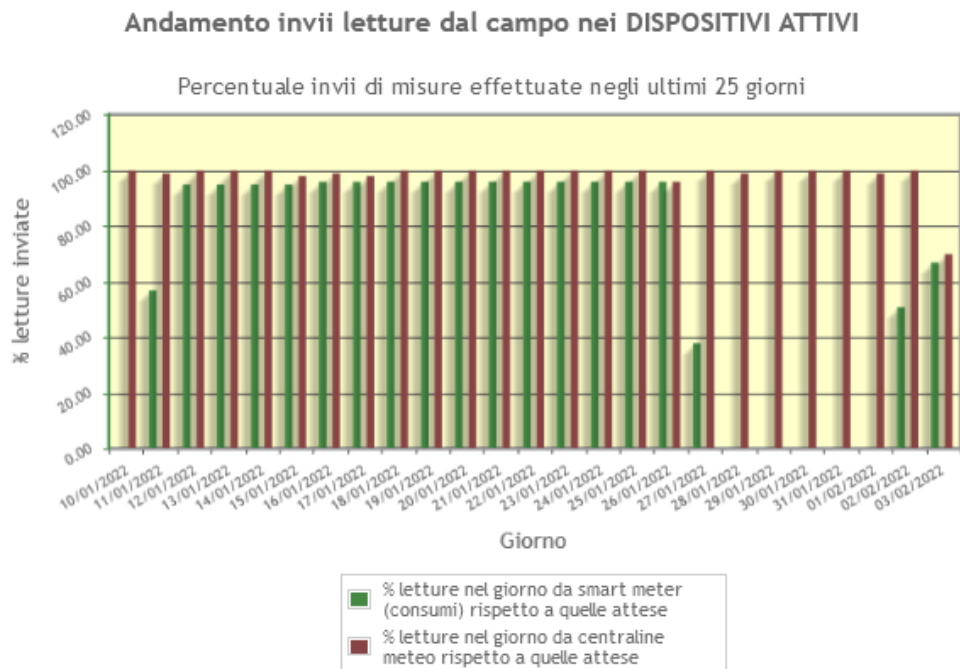


Figura 45: Esempio di monitoraggio del numero di misurazioni

1. Letture nel giorno da smart meter (consumi) rispetto a quelle attese (%)

Accesso: amministratore

Query:

```
SELECT *
FROM (select vws_scand_targetfrequency96normalised".*
      FROM "vws_scand_targetfrequency96normalised" LIMIT %%NumberofDays%%
      ) as orderedlist
ORDER BY orderedlist."timecomplete" ASC
```

Paramteri:

- %%NumberofDays%% identifica il numero di giorni su cui si vuole fare il monitoraggio.

2. Letture nel giorno da centraline meteo rispetto a quelle attese (%)

Accesso: amministratore

Query:

```
SELECT *
FROM (select "vws_scand_targetfrequencemeteo96normalised".*
      FROM "vws_scand_targetfrequencemeteo96normalised" LIMIT %%NumberofDays%%
      ) as orderedlist
ORDER BY orderedlist."time" ASC
```

Paramteri:

- %%NumberofDays%% identifica il numero di giorni su cui si vuole fare il monitoraggio.

5.4 Dati di irraggiamento solare

Altra componente necessaria per la gestione della Comunità Energetica è la misurazione dei rendimenti delle sorgenti energetiche, in particolare quelle fotovoltaiche che rappresenteranno, per il costo e la facilità di installazione, una delle maggiori sorgenti energetiche rinnovabili dei prossimi anni.

Per questo motivo, è necessario raccogliere informazioni di irraggiamento solare e potenza media generata dalle sorgenti a disposizione della Comunità. Ad esempio, in Figura 46 è riportato un grafico di dati di irraggiamento solare raccolti da un apposito sensore Tali dati sono confrontati con i dati di potenza raccolti da un pannello monocristallino.

Esempio di grafico per i dati di irraggiamento solare

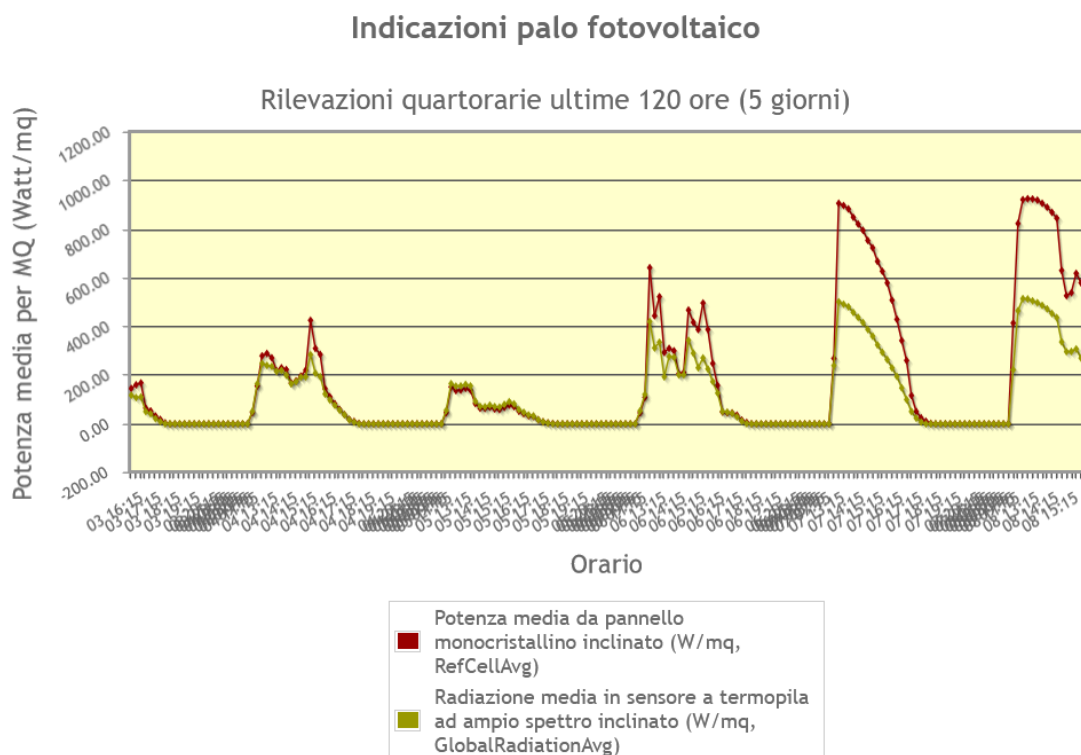


Figura 46: Esempio di grafico per i dati di irraggiamento solare

1. Potenza media da pannello monocristallino inclinato

La potenza media è misurata in W per unità di superficie, in modo da poter permettere un confronto omogeneo tra le differenti sorgenti, indipendentemente dalla superficie occupata dall'impianto. In questo modo, è possibile identificare immediatamente quegli impianti che producono quanto preventivato e quali, per un qualsiasi motivo, producono una quantità inferiore di potenza disponibile.

Equazione: Irraggiamento solare al giorno g , al tempo t su un piano inclinato: $I_{FV,g,t}$ [W/m²]

Accesso: utente

Query:

```
SELECT ctid, concat(
    "substring"(vw_palofv."timestamp"::character varying::text, 9, 2),
    ',
    "substring"(vw_palofv."timestamp"::character varying::text, 12, 5)) AS ascissa,
    vw_palofv."timestamp"::character varying AS dataora,
    vw_palofv.refcellavg AS potenzamedia,
    vw_palofv.airtempavg AS temperaturamedia,
    vw_palofv.globalradiationhoravg as radiazionemedi,
```

```
vw_palofv.deviceid,
row_number() OVER () AS ctid
FROM vw_palofv
WHERE vw_palofv.flowid = %%flowid%% and vw_palofv."timestamp">=NOW()
- %%interval%%':INTERVAL
ORDER BY vw_palofv."timestamp" ASC;
```

Paramteri:

- %%flowid%% identifica il flusso di dati in ingresso e, di fatto, una sorgente di potenza
- %%interval%% identifica un intervallo di tempo, in ore, per cui si vuole misurare l’irraggiamento solare.

5.5 Confronto tra produzione e consumo di energia

Elemento fondamentale sia in fase di progettazione che durante il funzionamento della Comunità Energetica è il confronto tra la produzione di energia ed il consumo di tutti gli utenti. Questo permette di calcolare le grandezze fondamentali per la Comunità, come la quantità di energia condivisa, la percentuale di autosufficienza energetica e il fabbisogno dalla rete. Questi dati permettono anche, ad esempio, di capire se l’investimenti in sistemi di accumulo o in ulteriori sorgenti è sostenibile e sensato per la comunità in esame.

La Figura 47 mostra il grafico di dati di produzione energetica da fotovoltaico confrontati con il consumo medio della comunità nello stesso periodo della produzione. Si riconosce una certa periodicità dei consumi di energia e la tipica campana della produzione da fotovoltaico. In questo grafico, poiché i dati di produzione da pannello fotovoltaico non sono disponibili, è stata usata una produzione stimata a partire dai dati di irraggiamento solare per la zona di Bologna.

Il grafico confronta i risultati di due operazioni:

- La stima della produzione in un certo periodo;
- I valori medi di consumo nello stesso periodo.

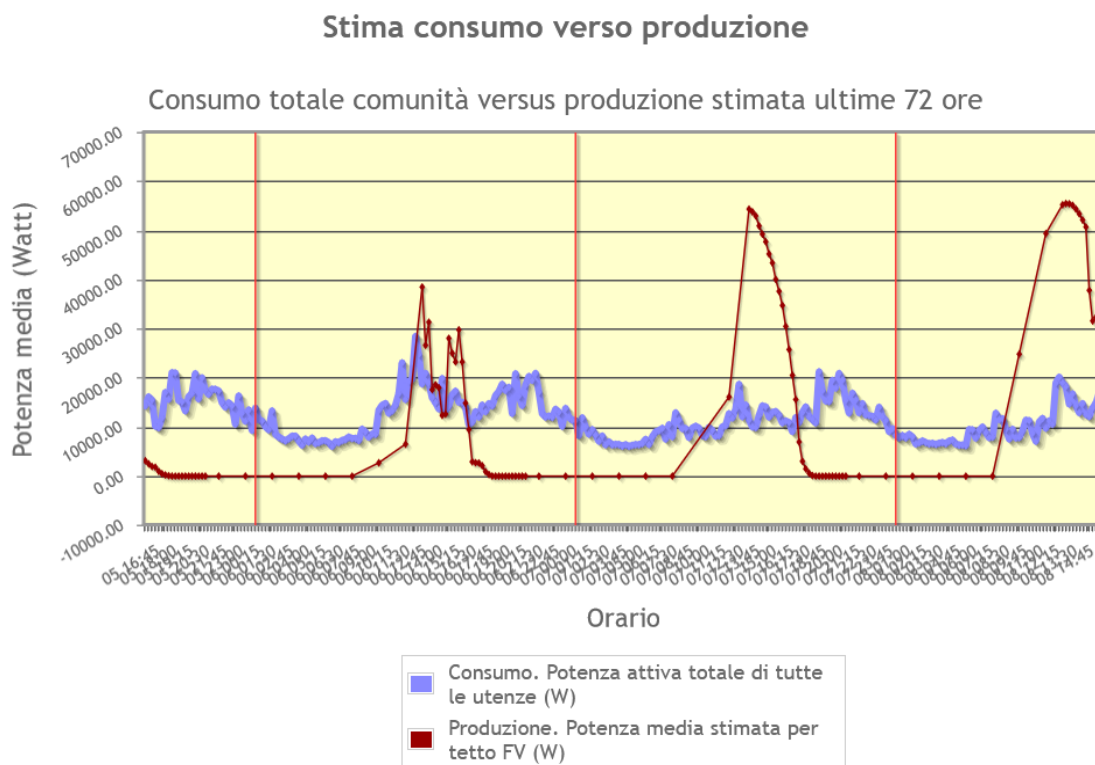


Figura 47: Esempio di grafico di produzione da FV

1. Produzione. Potenza media stimata per tetto FV

Equazione: Conteggio simulato della produzione dal tetto:

$$P_{FV}[W] = I_{FV,g,t} \cdot \xi \cdot S.$$

Dove $I_{FV,g,t}$ è il dato di irraggiamento definito precedentemente, ξ è l'efficienza dell'impianto FV (es. $\xi=0,1796$), S la superficie dell'impianto [m^2] (es. $S=334 m^2$).

Accesso: utente

Query:

```
SELECT
    ctid, concat(
        "substring"(vw_palofv."timestamp"::character varying::text, 9, 2),
        ',
        "substring"(vw_palofv."timestamp"::character varying::text, 12, 5)) AS ascissa,
    vw_palofv."timestamp"::character varying AS dataora,
    vw_palofv.refcellavg * %%conversionRate%% AS potenzamedia,
    vw_palofv.airtempavg AS temperaturamedia,
    vw_palofv.globalradiationhoravg as radiazionemedi,
    vw_palofv.deviceid,
    row_number() OVER () AS ctid
    FROM vw_palofv
        WHERE vw_palofv.flowid = %%flowid%%
        and "timestamp">=NOW() - (%%startHours%% hours)::INTERVAL
        and "timestamp"<NOW() - (%%endHours%% hours)::INTERVAL
    ORDER BY vw_palofv."timestamp" ASC;
```

Parametri

- %%flowid%% è il flusso che identifica il palo fotovoltaico che viene usato per la simulazione;
- %%startHours%% intervallo che si vuole graficare, ora di inizio dell'intervallo
- %%endHours%% intervallo che si vuole graficare, ora di inizio dell'intervallo, 0 per arrivare fino all'ultima lettura disponibile;
- %%conversionRate%% è un parametro di conversione, che dipende dal pannello fotovoltaico, tra l'irraggiamento e la quantità di potenza prodotta. Dipende dall'efficienza del pannello e dalla sua superficie.

5.6 Consumo mensile

Consumo mensile singolo utente

Per il consumo mensile (kWh/giorno) di uno specifico utente, si veda un esempio di grafico in Figura 48. I consumi vengono posti a zero quando la quantità di dati è inferiore al 90% di quelli previsti (sono previsti 96 dati al giorno moltiplicati per i giorni del mese), per evitare di mostrare dati non veritieri.

Consumo mensile dello specifico deviceid=5246

kWh di consumo ogni mese (sono stati posti a zero i punti aventi meno del 90% dei dati previsti)

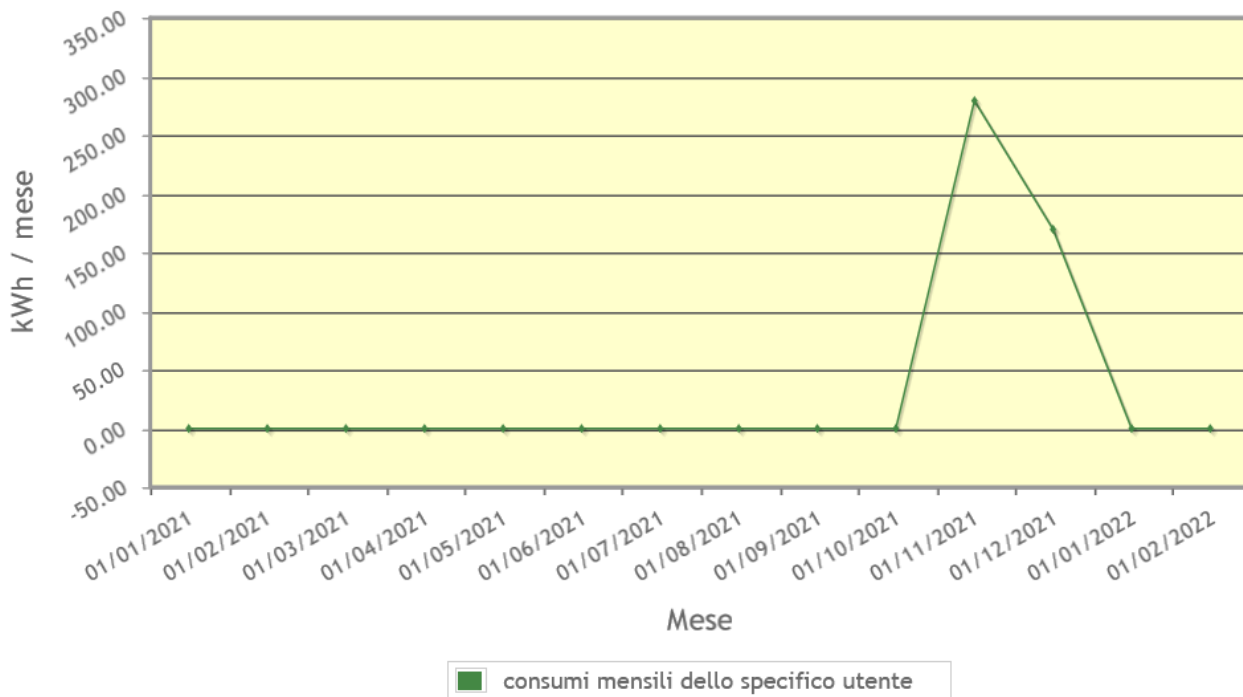


Figura 48: Esempio di consumo mensile per uno specifico utente

Equazione: Consumo nel mese dell'utente *i*=

$$C_{mese,i}[Wh] = \sum_{g=1}^{g_{mese}} \sum_{t=1}^{96} E_{i,g,t}$$

sommatoria nel tempo e da 1 ai giorni del mese (es. $g_{mese}=30$ per novembre).

Accesso: utente

Query:

```

SELECT *
FROM
    (SELECT ctid, time as timecomplete3,
    consumpmnth as kwh3,
    CASE WHEN precmonth >%%percMonth%% THEN consumpmnth
    ELSE 0
    END
    as consumpmnthprec
    FROM vw_consumption_month
    where deviceid='%%P2%%'
    LIMIT %%numeroMesi%%) as orderlist
ORDER BY orderlist."timecomplete3" ASC
    
```

Parametri:

- **percMonth** è un parametro che stabilisce la percentuale minima di dati al di sotto della quale il valor medio non viene calcolato
- **P2** - identifica l'utente che viene monitorato;
- **numeroMesi** - identifica il numero dei mesi che si vuole mostrare.

Consumo mensile del totale della comunità

Questa operazione restituisce un valore numerico unico per ogni mese per cui viene effettuato il calcolo.

Equazione: Consumo nel *mese* dell'intera comunità

$$C_{mese}[Wh] = \sum_{i=1}^N \sum_{g=1}^{g_{mese}} \sum_{t=1}^{96} E_{i,g,t}$$

sommatoria su tutti gli utenti *i*, sui giorni del mese (es. $g_{mese}=30$ per novembre), e sul tempo *t*.

Accesso: utente

Per il consumo mensile (kWh/giorno) dell'intera comunità, si riporta la seguente query. I consumi vengono sempre posti a zero quando la quantità di dati è inferiore al 90% di quelli previsti.

Query:

```
SELECT *
    FROM
    (SELECT ctid, time as timecomplete3,
    consumpmnth as kwh3,
    CASE WHEN precmonth >percMonth THEN consumpmnth
    ELSE 0
    END
    as consumpmnthprec
    FROM vw_consumption_month_all
    LIMIT numeroMesi) as orderlist
ORDER BY orderlist."timecomplete3" ASC
```

Parametri:

- **percMonth** è un parametro che stabilisce la percentuale minima di dati al di sotto della quale il valor medio non viene calcolato
- **numeroMesi** - identifica il numero dei mesi che si vuole mostrare.

5.7 Approccio alla visualizzazione dei dati

I dati raccolti permettono delle visualizzazioni e analisi di dettaglio che, normalmente, non sono utili per l'utente singolo e, in alcuni casi, fuorvianti o controproducenti.

In particolare, visualizzazioni di dati di dettaglio o molto complesse possono spaventare l'utente non opportunamente preparato che ha bisogno, invece, di pochi dati e immediatamente fruibili.

Ad esempio, un grafico come quello in Figura 47: Esempio di grafico di produzione da FV Figura 47 andrebbe illustrato e spiegato in dettaglio perché possa risultare utile ad un utente consumatore, mentre la Figura 42 risulta più facilmente leggibile.

Per questo motivo, si propone la visualizzazione dei dati attraverso una dashboard con 2 diversi livelli, uno più generale e uno più di dettaglio, da modulare a seconda che il riferimento sia l'utente non tecnico o un amministratore tecnico:

- Il primo livello con i parametri ritenuti più essenziali:
 - o Il carico di potenza [W] istantanea (es. ogni 15') dell'ultima settimana: $\sum_n C_n(t_i)$ dell'utente
 - o La curva di produzione totale [W] istantanea, se presente nella comunità, dell'ultima settimana, sovrapposta alla curva di carico totale: $\sum_i P_i(t_k)$.
 - o Consumo energetico [kWh] mensile e annuale totale dell'utente/comunità, evidenziando la variazione nel tempo. Confronto con la media della comunità e/o con media nazionale. Si

noti che il KPI kWh/anno/utenza deve almeno tenere conto del numero di residenti nell'abitazione e quindi il consumo deve essere confrontato con utenze con pari numero di componenti (non vi è semplice proporzione tra consumi e numero di componenti). Se non sono noti il numero di componenti della famiglia, è possibile mostrare una tabella con le medie dei consumi corrispondenti a 1,2,3,..n componenti. Sarà poi l'utente ad auto-valutare il proprio consumo.

- *Consumo energetico giornaliero* medio (o totale) [kWh] dell'utenza/comunità.
 - 2 cluster del *carico di potenza medio giornaliero* [W], uno per i giorni feriali e uno per quelli non feriali (distinti per stagionalità), con massimo, minimo, deviazione standard in ogni punto della giornata;
 - *Autosufficienza energetica* dell'utente/comunità [%] (rapporto tra autoconsumo e consumo) e *autoproduzione energetica* (rapporto tra autoconsumo e produzione) della comunità [%] e andamento nel tempo (variazione giornaliera);
 - Eventuali *token* guadagnati (nel caso di implementazione di premi in base a comportamenti virtuosi);
- Un secondo livello, accessibile attraverso link nelle diverse voci del primo livello, con informazioni più dettagliate (soprattutto per l'amministratore di comunità).
- Altre grandezze: *tensione* [V], *corrente* [A], *potenza reattiva*, *temperatura interna ed esterna* (se sono presenti sensori), *potenza accumulata* [kWh] (in caso di storage);
 - valori *massimo* e *minimo* totale raggiunto [W] nel giorno, mese e anno precedenti (per individuare la potenza massima da fornire ed il carico di base) dell'utente/comunità. Grafico della variazione del max/min nei diversi mesi;
 - *Ripartizione consumo energetico* [%] per fasce orarie f1, f2, f3 (normalizzato al numero di ore di ogni fascia);
 - *media carico di potenza* [W] in *cluster per giorni* (algoritmo che divide in maniera automatica ad es. comportamenti di lunedì/giovedì e venerdì/domenica) separati per ogni mese;
 - *media carico di potenza* [W] in *cluster per utenze "simili"*, similitudine può essere intesa per tipo di uso (residenze, esercizi commerciali, ecc.) o per tipo di curva;
 - *incidenza carico di base sul consumo totale* [%];
 - *Classifica dei consumi medi mensili* [kWh] degli utenti (per amministratori).

6 Conclusioni

Il lavoro di qualificazione dei servizi realizzati nel WP1 del programma sulla ricerca del sistema elettrico ha permesso di evidenziare le possibilità, ma anche le criticità, degli strumenti sviluppati a supporto dei differenti aspetti della Local Energy Community.

Differenti elementi critici, come la disponibilità dei dati di consumo e produzione o l'effettivo coinvolgimento dei cittadini all'interno delle iniziative locali rappresentano un ostacolo molto alto da superare in tutti i contesti. Nella gestione della propria richiesta di energia, questi problemi diventano ancora più rilevanti perché vanno ad impattare nella sfera privata delle persone e delle aziende. Infatti, i dati energetici, se letti correttamente, possono fornire informazioni rilevanti sulle abitudini delle persone o, nel caso di imprese, sulla produzione e sull'andamento economico dell'azienda stessa.

ENEA si propone di fornire servizi che possano rappresentare, nell'attuale mercato dell'offerta di energia e di servizi per le Comunità Energetiche, degli elementi di riferimento e comparazione per tutti gli operatori di mercato con l'obiettivo specifico di non turbare il mercato o di competere con aziende private fornitrici di servizi.

Gli strumenti presentati coprono la maggior parte del ciclo di vita della Comunità Energetica e guardano verso l'evoluzione "smart" delle stesse in modo che l'energia diventi un punto di accesso e di condivisione di una lunga lista di beni e servizi che possano valorizzare il territorio.

Tali strumenti, anche se per la maggior parte non sono pronti per essere resi pubblici e garantire un livello adeguato di servizio, rappresentano lo stato dell'arte nello sviluppo degli strumenti a supporto delle Comunità Energetiche e hanno generato, quando presentati durante le fasi di disseminazione e presentazione, un interesse molto alto in differenti operatori del settore.

L'evoluzione delle attività nell'ultimo periodo dell'attuale progetto dovrà concentrarsi in particolare su:

- Testare i vari elementi singolarmente per identificare eventuali problematiche;
- Armonizzare la collaborazione tra i vari servizi sviluppati;
- Testare gli strumenti integrati.

Riferimenti bibliografici

- [1] I. Ajzen (1991). The theory of planned behavior. *Org. Behav. Human Decision Process.*, vol. 50, no. 2, pp. 179-211. DOI: 10.1016/0749-5978(91)90020-T.
- [2] I. Ajzen (2002). Perceived behavioral control, self-efficacy, locus of control, and the theory of planned behavior. *J. Appl. Social Psychol.*, vol. 32, no. 4, pp. 665-683. DOI: 10.1111/j.1559-1816.2002.tb00236.x.
- [3] D.L. Schiera, F.D. Minuto, L. Bottaccioli, R. Borchiellini, A. Lanzini (2019). Analysis of Rooftop Photovoltaics Diffusion in Energy Community Buildings by a Novel GIS- and Agent-Based Modeling Co-Simulation Platform". *IEEE*. DOI: 10.1109/ACCESS.2019.2927446
- [4] E. M. Rogers (2003). *Diffusion of Innovations*. 5th ed. New York, NY, USA, Free Press.
- [5] G. Deffuant, F. Amblard, G. Weisbuch, and T. Faure (2002). How can extremism prevail? A study based on the relative agreement interaction model. *J. Artif. Soc. Social Simul.*, vol. 5, no. 4. [Online] <http://jasss.soc.surrey.ac.uk/5/4/1.html>
- [6] <https://github.com/projectmesa/mesa/>