An Industrial Edge-Cloud Continuum Platform for Renewable Energy Communities

Carmine Colarusso[1], Assunta De Caro[1], Liudmila Dobriakova[2], Pio Antonio Perugini[1], and Eugenio Zimeo[1, 3]

[1]*Dept. of Engineering*, Università degli Studi del Sannio, Benevento, Italy
[2]Concept Reply, Italy
[3]*CINI Smart Cities and Communities Lab*

***Abstract*—Renewable Energy Communities (REC) enhance sustainability by aligning member consumption with renewable generation. This work introduces the concrete architectural implementation and industrial transfer of SOLAR, a REC management platform grounded in the Behavior-enabled IoT (BeT) paradigm. The proposed solution distributes core functionalities, such as monitoring, optimization planning and execution, and transparency management across an adaptive Edge-Cloud Continuum centered around a Community Digital Twin.**

***Index Terms*—Energy Management, Renewable Energy Communities, Behavior-enabled IoT, Digital Twin, Edge-Cloud architecture**

I. Introduction

A Renewable Energy Community (REC) is a legal framework that enables consumers, prosumers, and producers to collaborate to generate, distribute, and consume green energy. Its primary goal is to deliver environmental, social, and economic benefits to its members and the local area, largely through incentives that reward maximizing self-consumption. Under Italy's virtual self-consumption model [1], shared energy is calculated hourly as the minimum between the renewable energy injected into the grid by the REC and the total consumption within the same grid section. Within this legal framework, the community's members are financially compensated for this shared volume through a combined valorization component and a premium incentive tariff. [2]. Therefore, to exploit REC's full potential, the maximization of virtual self-consumption is required, that is, aligning REC's withdrawn energy with the energy produced. Even though maximizing economic profit is the main objective of REC's members, their comfort needs to be considered to build sustainable communities. While conventional REC management solutions focus strictly on load optimization or lack structured frameworks to balance technical efficiency with user comfort, our approach shifts the focus toward a user-centric model. Specifically, the platform is grounded in the Behavior-enabled IoT (BeT) paradigm [3], that systematically manages the complex trade-offs between system-level Quality of Service (QoS) and user-perceived Quality of Experience (QoE) through a bi-causal loop. Grounded in this framework, we previously introduced SOLAR (Self-Optimizing REC management pLAtfoRm) [4], a BeT-driven architecture that finds a Pareto-optimal solution by scheduling home appliances and community battery energy storage systems (BESSes). Moving beyond that theoretical foundation, this work tackles the concrete challenges of transferring and industrializing the SOLAR platform within the real-world IoT4EnergyCommunities project.

This work has been partially supported by the IoT4EnergyCommunities project founded by MIMIT Prog n. FTE0000482.

II. REC Management Architecture

To concurrently balance user comfort and REC efficiency, we propose a platform relying on a hybrid Edge–Cloud topology aligned with the functional architecture shown in Figure 1. Local data acquisition, filtering, and real-time automation are offloaded to Edge nodes, while the Cloud layer (mainly implemented with Microsoft Azure technologies) coordinates wide-area optimization, predictive modeling, and analytics.

In particular, Smart meters distributed across REC assets (e.g., community BESSes, generation facilities, and consumer premises), generate telemetry streams at multiple granularities, spanning from appliance-specific readings to building-level aggregations. Because these monitoring devices are heterogeneous, vendor-specific IoT systems, the W3C Web of Things standard is adopted to harmonize device profiles and data interfaces. Once standardized, these streams are routed through an Azure IoT Hub for secure ingestion and forwarded via an event-driven Azure Event Grid to serverless Azure Functions. These functions dynamically parse JSON payloads based on device type to update the Community Digital Twin (DT). This last is implemented via Azure Digital Twins and modeled through the Digital Twins Definition Language (DTDL).

The DT serves as an event-driven semantic knowledge layer mapping real-time data onto a structured graph to underpin monitoring and diagnostics, while its historical data enables the reconstruction of energy dynamics and behavioral patterns. Furthermore, for devices that expose remote actuation capabilities, the DT and IoT Hub could dispatch control commands in the reverse direction, closing the bidirectional control loop.

Historical data managed by the DT is essential for the Community Planner, who computes day-ahead or online scheduling plans via three core modules. *i)* The Preference Model captures user energy requirements and daily habits (explicit or inferred) to define temporal boundaries as QoE constraints. To model and forecast multi-step household and community demand, the module implements an XGBoost-based Multi-Output Regressor. Specifically, a separate estimator is trained for each of the 168 forecasting horizons. The model uses mean squared error

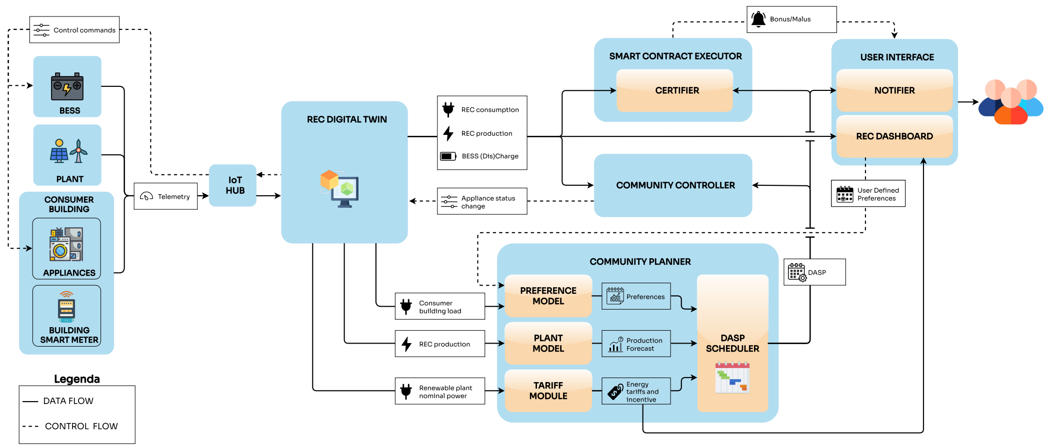

Fig. 1. BeT-driven REC management architecture. Solid and dashed lines represent the data and control flow, respectively.

loss and histogram-based tree construction, configured with 1000 boosting stages, a 0.03 learning rate, a maximum depth of 6, and 0.8 subsampling and column sampling factors. *ii)* The Plant Model is a supervised multi-horizon forecasting problem using an XGBoost-based Multi-Output Regressor wrapper. The input features are engineered via a sliding window to capture historical and exogenous dynamics, and independent estimators are trained at each horizon step to model step-specific nonlinear patterns efficiently. To reduce computational overhead, hyperparameter tuning via Optuna minimizes the average RMSE sampled every 4 hours across a subset of horizons, searching over tree depth, learning rate, subsampling ratios, and regularization. The final model is evaluated across the full 168-hour horizon using multiple standard metrics (RMSE, MAE, MAPE, R^2, and a persistence skill score). Gain-based feature importance, averaged across all estimators, reveals that recent production history dominates, followed by temperature dynamics and cyclic temporal features. Finally, *iii)* the Tariff Module integrates regulations, incentives, and certified energy flows to drive economic efficiency. By synthesizing these inputs, a platform Scheduler generates an operational plan to coordinate energy optimization across all REC assets. The scheduler solves an optimization problem formulated as a Mixed-Integer Linear Programming model implemented in Python using the PuLP library. The model is executed programmatically via the Coin-or Branch-and-Cut solver; however, to ensure system scalability during large-scale operations, an automatic fallback to a heuristic Greedy algorithm is enforced.

Together with Community Planner, the Community Controller executes the plan across the target temporal horizon (e.g., daily) by updating the DT's state variables, which are dispatched as physical commands via the IoT Hub. Real-time monitoring mechanisms continuously compare observed consumption and storage states with the reference schedule, enabling detection of deviations caused by stochastic environmental conditions or user-driven behavioral shifts. When significant deviations are identified, the system triggers adaptive re-optimization procedures to maintain operational feasibility and uphold QoS and QoE constraints.

Finally, a dedicated User Interface (UI) drives member engagement by exposing a community dashboard (aggregating cumulative production, consumption, and economic benefits) and a personalized view (tracking individual contributions and plan adherence). Moreover, the UI integrates a notification system that alerts members when a plan is published, dispatching tailored appliance activation prompts.

In summary, the platform's logical architecture naturally maps to the autonomic MAPE-K loop: Monitoring (IoT Hub/telemetry), Analysis (Community DT), Planning (Scheduler), Execution (Community Controller), and Knowledge (DT/Certifier). To transition towards a probabilistic framework, the implementation of the Edge–Cloud architecture is envisioned as follows. The functional components can be deployed as containerized microservices across an adaptive Edge-Cloud Continuum to balance latency and resilience: K3s-managed Edge nodes handle low-latency ingestion and local automation, while Cloud infrastructure hosts intensive optimization, Delta Lakes, and the Community DT. To ensure transparency, Hyperledger Fabric-based Certifier immutably logs telemetry and scheduling plans, leveraging a permissioned blockchain topology to ensure regulatory compliance and data governance. Finally, a Smart Contract Executor programmatically manages member compliance and behavioral token distribution by cross-referencing real-time energy data with the certified plans. This direct feedback loop could reinforce participant awareness and incentivize compliant, grid-virtuous behavior.

References

[1] ARERA, "Definizione, ai sensi del decreto legislativo 199/21 e del decreto legislativo 210/21, della regolazione dell'autoconsumo diffuso. approvazione del testo integrato dell'autoconsumo diffuso," 2022.

[2] Gestore Servizi Energetici, "DECRETO CACER e TIAD–regole operative per l'accesso al servizio per l'autoconsumo diffuso e al contributo PNRR," 2024.

[3] H. Muccini, B. Russo, and E. Zimeo, "The bet project: Behavior-enabled iot.," *arXiv*, 2023.

[4] A. De Caro and E. Zimeo, "Behavior enabled iot: A software architecture for self-adapting a renewable energy community," *Internet of Things*, vol. 27, p. 101294, Oct. 2024.