A Work Project, presented as part of the requirements for the Award of a Master's degree in
Management from the Nova School of Business and Economics

Assessing the Impact of Peer-to-Peer Energy Trading on Financial Benefits and Autonomy in Local Communities using an Agent-Based Model

Fabian Tim Simon – 58932

Work project carried out under the supervision of:
Leid Zejnilovic

Defense: 21st January 2025

16/02/2025

Abstract

This thesis explores the transformative potential of Peer-to-Peer (P2P) energy trading as an extension to existing centralized systems, where energy is distributed exclusively by the utility company. Using agent-based modeling (ABM), the study simulates interactions between prosumers, consumers, and utility providers to examine the benefits of P2P trading. The findings reveal that P2P trading provides financial benefits, enhances energy autonomy by enabling local energy use, and improves energy resilience by decreasing reliance on centralized grids. These insights provide actionable recommendations for end-users, policymakers, and utilities, highlighting the role of P2P systems in advancing a decentralized and consumer-driven energy transition.

Keywords: Peer-to-Peer Energy Trading, Decentralized Energy Systems, Energy Autonomy, Cost Savings, Dynamic Pricing, Sustainable Communities

This work used infrastructure and resources funded by Fundação para a Ciência e a Tecnologia (UID/ECO/00124/2013, UID/ECO/00124/2019 and Social Sciences DataLab, Project 22209), POR Lisboa (LISBOA-01-0145-FEDER-007722 and Social Sciences DataLab, Project 22209) and POR Norte (Social Sciences DataLab, Project 22209).

Table of Contents

Table of Figures

Group component

Introduction

The energy industry is undergoing a transformative shift toward more sustainable production practices. Households increasingly play an active role in this transition by generating their own energy through distributed energy resources (DERs). The rapidly declining costs of DERs, such as solar panels, have enabled households to effectively reduce energy expenses while decreasing reliance on centralized grid systems. Despite the widespread adoption of residential photovoltaic systems in developed countries, the inherent cyclic nature of solar energy production poses a significant challenge. In response, Peer-to-Peer (P2P) energy trading within local communities has emerged as a promising solution. Households with surplus energy can trade it with those facing a deficit, enabling prosumers, households that produce and consume energy, to actively participate in energy markets. Local energy sharing can lower costs while enhancing community energy autonomy and resilience. Real-world case studies, such as the Brooklyn Microgrid project, highlight the potential of P2P trading in transforming energy systems. Moreover, advancements in blockchain technology have emerged as a powerful tool enabling energy trading by providing a secure and transparent way for households to exchange energy directly. A decentralized system reduces the need for intermediaries, fostering more efficient local energy trades and empowering communities to manage their energy use independently. By supporting trust and automation, blockchain can create resilient, cost-effective energy networks.

Despite the rising prominence of P2P energy trading facilitated by blockchain technology, there is a gap in understanding how these systems impact end-users and local communities in practical terms. While existing research often pursues rather theoretical approaches, for instance, by applying game-theoretic models, few studies examine the tangible benefits for households participating in P2P trading. Therefore, delivering quantifiable results is essential to motivate more households to engage in local energy trading. Addressing this gap is vital, as the widespread adoption and effectiveness of P2P energy trading depend on its ability to provide meaningful value for end-users. Although the model presented in this thesis studies outcomes from an end-user perspective, the findings offer actionable insights for a broader range of stakeholders, including policymakers and energy providers.

This thesis aims to contribute to the existing literature by simulating energy trading between households using data from the German state of Baden-Württemberg. An agent-based model examines outcomes under realistic conditions, making the findings applicable beyond the specific geographic scope of the region. A base scenario, representing a system without P2P trading under an active feed-in tariff, is compared to alternative scenarios where households are enabled to exchange energy. The analysis investigates financial benefits, the sum of consumer Cost Savings and prosumer P2P Revenue, and energy autonomy from an end-user perspective while also exploring the role of dynamic pricing in facilitating P2P trading.

Literature Review

With the rise in energy production by private households, local energy trading has become an increasingly popular topic among researchers. Andoni et al. (2019) emphasize that P2P energy trading offers prosumers an additional revenue source, leading to lower energy costs. Moreover, it helps reduce the physical load on national grid systems and optimizes energy flow at the community level (Andoni et al., 2019). Given that the EU requires an estimated €200 billion annually to transition toward a more sustainable and secure energy infrastructure, local energy production and consumption offer a promising way to reduce reliance on long-distance energy transmission (Andoni et al., 2019). This literature review begins by exploring key contributions to P2P energy trading, examines the use of agent-based models, investigates the role of blockchain technology in facilitating these systems, and concludes with real-world case studies of energy trading communities.

Peer-to-Peer Energy Trading

The Peer-to-Peer (P2P) energy trading literature has extensively focused on optimizing trading outcomes, often by applying linear programming approaches. Nguyen et al. (2018) simulate interactions where households can trade self-produced surplus energy with peers under real-world constraints, such as demand, energy production, and battery storage limitations. Their findings highlight that outcomes are particularly sensitive to the amount of self-produced energy, the number of prosumers, and battery storage capacity. They conclude that prosumers with sufficient battery storage and photovoltaic (PV) systems can achieve up to 28% higher financial savings than those under an active feed-in tariff. Similarly, Alam et al. (2019) examine P2P energy trading within small communities from an end-user perspective. Their research reveals that cost savings are often shared unequally among households. To address this, the authors propose a bi-linear programming approach to develop a trading scheme that ensures Pareto-optimal outcomes, where no household can improve its situation without making another worse off. They also observe that cost savings do not increase linearly with the share of prosumers or storage capacity, instead maximum savings are achieved at a specific saturation point.

Another area of research explores the role of pricing in P2P energy trading. Liu et al. (2017) present a pricing model to encourage interactions in a community of prosumers. Their study demonstrates that households gain greater financial benefits from P2P trading compared to relying on utility companies. Wu et al. (2018) investigate P2P energy trading in energy networks by comparing two pricing strategies for their ability to maximize profits for prosumer households. Their results indicate that P2P trading can reduce electricity prices by 5% to 15%.

Additionally, they suggest that a differentiated pricing strategy is more effective than a unified approach in incentivizing prosumers to trade energy due to higher revenues.

A further stream of literature examines the application of game theory to P2P energy trading. Much of this research focuses on finding stable trading coalitions and equilibria while considering motivational and psychological factors. Tushar et al. (2019) propose a motivational framework to enhance prosumer participation in P2P energy trading. They conclude that consistent participation of prosumers is crucial for P2P trading efficiency, emphasizing the importance of motivational factors. Zhang et al. (2019) use Elecbay, an online energy trading platform, to simulate P2P bidding using game theory. Their findings demonstrate that P2P trading shifts prosumer load profiles and improves supply-demand matching within local communities. These results are validated in a case study using data from the European Union Benchmark Microgrid Network (Zhang et al., 2019). Long et al. (2019) emphasize the need to motivate prosumers to fully unlock the benefits of P2P trading. They propose a fairness-focused trading mechanism that optimally benefits prosumers.

Agent-Based Models in P2P Energy Trading

Agent-based models (ABMs) constitute a smaller subset of the literature on P2P energy trading, with the majority of research favoring methods like game theory or optimization approaches. However, ABMs offer unique advantages by capturing complex interactions between agents and uncovering emergent behaviors, making them valuable for studying decentralized energy systems and community dynamics. Guimaraes et al. (2021) use an ABM to explore P2P energy trading in a localized grid environment. Their study simulates interactions among 100 American households, using real-world data to model energy consumption and self-production. By varying the proportion of prosumers in the simulated community, they observe that trading activity peaks when prosumers comprise 50–75% of the households. Additionally, they highlight that grid dependence is significantly reduced, and the system's overall resilience is strengthened even with a small share of prosumers. The findings suggest that increasing the proportion of prosumers within a community can lead to more balanced energy flows and enhanced network stability, emphasizing the importance of prosumer participation in achieving optimal outcomes. Monroe et al. (2020) adopt a slightly different approach, using an ABM to simulate P2P trading while directly comparing the results to a real-world case study conducted in Perth, Australia. Their research highlights the role of energy storage infrastructure on market performance and energy pricing. They demonstrate that sufficient battery storage increases community autonomy and effectively mitigates demand peaks. By showing how localized energy storage can stabilize energy systems, Monroe et al. highlight the potential for ABMs to address both technical and economic aspects of energy trading systems.

Despite these contributions, certain aspects of P2P energy trading still need to be explored within the ABM literature. For example, the role of dynamic pricing mechanisms, which could better align economic incentives between prosumers and consumers, has received limited attention. Furthermore, limited research has explored how cost savings and additional prosumer revenue are distributed within localized networks, a central aspect for understanding the financial dynamics among stakeholders. Addressing these gaps would not only deepen the understanding of P2P energy trading dynamics but also expand the applicability of ABMs in designing efficient energy trading systems.

The Role of Blockchain in P2P Energy Trading

To enable effective energy exchange among households, the underlying technology must meet critical requirements such as security and transparency. By addressing these needs, Blockchain has emerged as a promising solution for P2P energy trading. This section evaluates the benefits, challenges, and current applications of blockchain technology in P2P energy trading. Blockchain is a decentralized, distributed digital ledger that securely records transactions across a network of computers without requiring a central authority (Wongthongtham et al., 2021).

Transactions are stored in cryptographically linked blocks, forming an immutable chain that ensures data integrity, resilience, and transparency, key attributes for building trust in P2P trading systems (Cali & Fifield, 2019). A major advantage of blockchain is its ability to automate and facilitate transactions through smart contracts. These self-executing contracts enable prosumers to sell surplus energy directly to consumers by automating agreements and executing transactions once predefined conditions are met (Cali & Fifield, 2019). This eliminates the need for intermediaries, reducing operational costs and improving efficiency (Evens et al., 2023). Furthermore, the dynamic markets enabled by blockchain-powered auctions and real-time energy pricing further benefit prosumers and consumers by promoting cost-effective energy trading (Evens et al., 2023). Blockchain's ability to eliminate intermediaries and provide tamper-proof transactions empowers consumers and prosumers, encouraging active market participation (Andoni et al., 2019).

Despite its potential, blockchain technology faces several challenges. Scalability issues, high energy usage, and development costs limit its adoption, particularly in large-scale energy trading systems where efficiency and speed are essential (Andoni et al., 2019). Regulatory uncertainties and the lack of interoperability between blockchain networks further complicate implementation (Wongthongtham et al., 2021). Traditional databases may still be cost-effective alternatives for smaller-scale applications, although they lack Blockchain's transparency and immutability (Andoni et al., 2019). Innovations such as layer-2 scaling techniques and hybrid blockchain models are being developed to address these limitations, aiming to improve transaction speed, reduce energy consumption, and enhance the feasibility of blockchain for broader applications (Cali & Fifield, 2019). Andoni et al. (2019) highlight various use cases for blockchain in the energy sector, whereas this thesis focuses specifically on blockchain's role in facilitating P2P energy trading in localized, decentralized communities.

Case Studies on P2P Trading in Local Communities

Andoni et al. (2019) provide a comprehensive overview of blockchain's application in the energy sector, identifying over 140 blockchain-driven innovation projects and research initiatives. Prominent examples include the SonnenCommunity in Germany, Power Ledger in Australia, and the Brooklyn Microgrid in the United States. These projects utilize the blockchain to support localized energy trading, demonstrating benefits such as improved energy allocation, reduced participant costs, and enhanced system resilience (Cali & Fifield, 2019). The SonnenCommunity in Germany leverages blockchain to create a virtual network where members can trade excess solar energy. Participants with PV contribute their surplus energy to the community, while others draw from the shared pool as needed. This system promotes efficient energy use and fosters greater energy independence. Similarly, Power Ledger in Australia utilizes blockchain to enable real-time energy trading among households and businesses. Its platform provides financial incentives for participants by allowing them to set prices and trade energy directly, bypassing traditional intermediaries. Power Ledger's system also supports energy trading within microgrids and offers solutions for tracking renewable energy certificates. This flexibility has made them pioneers in demonstrating how Blockchain can support localized trading and broader energy market applications. The Brooklyn Microgrid in the United States is another widely cited example. This pilot project enables residents in a Brooklyn neighborhood to trade excess solar energy directly with one another. Using blockchain, participants can conduct secure, transparent transactions while maintaining full control over their energy data. The project has successfully demonstrated how blockchain can enhance community energy autonomy and resilience by creating localized energy markets. By optimizing the use of locally produced renewable energy, the Brooklyn Microgrid reduces reliance on centralized grids (Mengelkamp et al., 2017).

While these case studies highlight blockchain's potential to revolutionize energy systems, their success relies on several critical factors. Active community involvement is essential to ensure user engagement and trust in these systems. Supportive legal and regulatory frameworks are necessary to legitimize blockchain-enabled trading and address data privacy and compliance issues. Additionally, the widespread adoption of blockchain technology requires addressing barriers such as high upfront costs, scalability limitations, and ensuring equitable access for all participants (Evens et al., 2023; Umar et al., 2024). These challenges also highlight the importance of designing systems that balance technological innovation with community needs.

Limitations & Future Research

While this study provides valuable insights into P2P energy trading systems, several limitations must be acknowledged to contextualize the findings and identify areas for future research.

Methodological Limitations

Agent-based modeling is a powerful tool for simulating decentralized energy systems yet involves certain trade-offs. ABM simplifies real-world complexity, and while the simulation's assumptions are grounded in existing data and literature, they may still limit the model's predictive accuracy. For instance, the model assumes that the utility company's pricing remains static and does not adjust in response to the emerging P2P market. In reality, utilities may adapt pricing strategies, potentially influencing consumer and prosumer behavior. While the current simulation cannot capture scenarios where the utility company dynamically adjusts its pricing, it could be extended in the future to incorporate such changes and provide insights into these interactions. Moreover, in AS1, the fixed P2P price of €0.24/kWh and the willingness-to-sell threshold in AS2 were chosen as the fair midpoints within the price range. These values were chosen as practical approximations for the simulation but do not account for individual agent preferences, regional price variability, or external economic influences, which could affect trading dynamics. Behavioral heterogeneity among agents, including different levels of risk aversion or preferences for renewable energy, could be subject to further research.

Another point to acknowledge is that the ABM does not account for transaction costs in either traditional energy trading or blockchain-based P2P trading. These costs include blockchain transaction fees and administrative and operational expenses in traditional systems, which could influence trading outcomes. Such costs were excluded from the model because they vary significantly depending on system setup, market structure, and regional policies. By omitting these costs, the findings are intended to be applicable across a broader range of scenarios.

Furthermore, the model assumes homogeneity in technological adoption and capacity among end-users. In reality, household consumption can vary widely between energy production, storage capacity, and consumption habits. Incorporating more detailed and diverse data in future simulations could help capture these differences, leading to a better understanding of how variations in technology affect trading patterns and outcomes.

Implementation Challenges

Implementing P2P trading systems faces a range of regulatory, technical, and economic challenges that must be addressed to ensure their feasibility and scalability.

Regulatory constraints represent a significant hurdle, as existing energy policies are often designed to support centralized grid models rather than decentralized systems. Adapting these frameworks to accommodate P2P trading will require targeted policy reforms, including standardized guidelines for energy transactions, clarity on legal liabilities, and mechanisms to integrate decentralized technologies into broader grid infrastructure. Such reforms are necessary for the full potential of P2P trading systems to remain realized.

From a technical perspective, creating a secure and efficient marketplace for P2P trading is critical. Blockchain technology has emerged as a promising solution to ensure participant transparency, security, and trust. However, scaling blockchain systems involves challenges such as integrating with existing grid infrastructure, handling the high computational demands of verifying transactions, and ensuring scalability for large numbers of users. Furthermore, ensuring interoperability between various blockchain platforms and energy management systems is essential to facilitate seamless energy trading. Addressing these challenges will require collaboration between technology developers, policymakers, and energy providers to ensure effective implementation.

Moreover, economic factors play a significant role in adopting P2P trading systems. High capital costs associated with PV systems, battery storage, and enabling technologies like smart meters may deter widespread participation. For lower-income households, financial barriers could prevent access to decentralized systems, raising concerns about equity and inclusivity. This disparity risks creating an energy divide, where the benefits of P2P trading are disproportionately accessible to wealthier households. Overcoming these barriers will require new financing options and collaborative approaches to make decentralized energy solutions accessible and beneficial for all socioeconomic groups.

Future Research Directions

To address these limitations and challenges, future research should explore key areas that can enhance the understanding and application of P2P energy trading systems.

Behavioral dynamics represent a critical area of exploration. Understanding the decision-making processes of consumers and prosumers can provide deeper insights into their trading behaviors. Risk tolerance, economic preferences, environmental awareness, and social influences will likely shape how agents interact within decentralized markets. Incorporating these elements into future simulations could refine predictions and identify strategies to encourage participation. Surveys, experiments, and real-world case studies could be used to gather data on these behaviors, complementing simulation-based approaches.

Another important research direction is the development of advanced forecasting and decision-making tools. Integrating predictive models for energy production and consumption and incorporating real-time data analytics can improve the accuracy of trading decisions. For instance, prosumers could use machine learning algorithms to predict future energy prices or production levels, optimizing the timing and volume of their trades. Enhanced forecasting capabilities would improve individual outcomes and contribute to overall system efficiency and stability.

Furthermore, research can examine the broader socioeconomic impacts of P2P trading systems by incorporating social dynamics into ABM simulations. Social norms and trust play a critical role in shaping how agents trade energy, influencing decisions about whether to participate, how much energy to share, and with whom. For example, trust in transaction reliability and pricing transparency can boost participation, while norms around sustainability and community cooperation may encourage more collaborative trading behaviors. Integrating these dynamics allows ABM simulations to provide deeper insights into how social factors drive adoption and impact system efficiency, equity, and accessibility.

Finally, future research should focus on designing and evaluating policy frameworks that support the widespread adoption of P2P trading systems. Policymakers must develop regulations that balance the interests of prosumers, consumers, and traditional energy providers while promoting equitable access to decentralized energy systems. This includes mechanisms to ensure grid stability, protect consumer rights, and incentivize renewable energy adoption. Comparative studies of regulatory approaches in different countries or regions could offer valuable insights into best practices and potential pitfalls.

Conclusion

This thesis examined the potential of P2P energy trading to enhance energy autonomy and financial benefits for end-users in decentralized energy systems. Using Agent-Based Modeling provided insights into the interactions between prosumers and consumers under different trading scenarios. The findings confirm that P2P energy trading can significantly improve community energy autonomy and offer economic incentives to prosumers and consumers. Dynamic pricing mechanisms were particularly effective in aligning economic interests and encouraging participation, highlighting the role of market-based solutions in advancing decentralized energy systems. However, challenges such as supply-demand mismatches and underutilized surplus energy emphasize the need for integrated solutions like community storage and demand-side management to realize the full benefits of P2P systems. These results carry important implications for stakeholders. Policymakers can leverage P2P trading to support renewable energy adoption, enhance energy resilience, and reduce grid dependency. Adapting to decentralized systems requires energy providers to rethink their role and develop strategies to respond to the emergence of P2P energy markets.

In summary, P2P energy trading is not just a theoretical concept but a practical solution for reshaping energy systems. By empowering communities to achieve greater energy autonomy and financial benefits, it challenges the reliance on traditional energy providers. P2P energy trading stands as a transformative approach to building sustainable, resilient, and community-driven energy systems, driving the shift toward a decentralized energy future.

Individual contribution

Methodology

Agent-Based Modeling (ABM)

Introduction to Agent-Based Modeling

Agent-based modeling (ABM) is a computer simulation technique designed to model complex systems by capturing autonomous agents' behaviors and interactions with their environment. It allows us to analyze how the collective dynamics of a system emerge from the decisions and interactions of its individual agents (Guimarães et al., 2021). In this context, agents represent individual entities, such as households (prosumers and consumers) or energy providers, who act according to predefined rules and are assigned certain attributes that influence the broader system (Guimarães et al., 2021). ABM is particularly useful because it accommodates heterogeneity among agents, incorporates adaptive decision-making, and can simulate non-linear interactions. According to Akhatova et al. (2022) and Yao et al. (2023), ABM is particularly well suited for studying dynamic and decentralized ecosystems, where traditional modeling approaches may fall short.

Rationale for using ABM in Energy Trading Systems

ABM has emerged as a valuable tool in energy system research for examining behavioral patterns and the interplay between social and technological changes. This approach is particularly relevant as modern energy systems grow increasingly complex, involving diverse agents such as prosumers, consumers, and utility providers, each with unique characteristics and interactions. Akhatova et al. (2022) argue that ABM is well-suited for capturing micro-level interactions and their macro-level consequences, especially in decentralized energy trading environments. By contrast, traditional equation-based models lack the flexibility to account for agents' heterogeneous and adaptive nature, making them less effective in analyzing decentralized systems (Akhatova et al., 2022).

We decided to employ an ABM to study P2P energy trading due to its capability to simulate complex real-world scenarios that are challenging to implement practically. Our ABM captures agents' decision-making processes by incorporating specific rules for trading behavior and battery storage utilization, which is essential for simulating realistic energy dynamics. These rules govern how prosumers prioritize energy use, trade surplus, and manage storage under different conditions. This approach enables a detailed analysis of localized energy trading. Moreover, the model offers various stakeholders insightful information by simulating different "what-if" scenarios and incorporating their perspectives. Additionally, when supported by empirical data, ABMs align closely with real-world scenarios, enhancing their predictive power and applicability to complex energy systems (Akhatova et al., 2022).

ABM Platform Selection: NetLogo

We selected NetLogo as the platform to implement our ABM simulation due to its robust functionality, extensive interface components, and user-friendly design. NetLogo is built on Scala and Java and has a strong reputation in academic research for its reliability and ease of use. It is particularly valued for modeling complex systems, including energy systems, making it well-suited for our simulation needs. The platform’s high degree of flexibility allows it to accurately capture the complexity of agent interactions across various levels of detail and scenarios. Additionally, NetLogo offers a graphical environment (3D View) that visually represents the simulation, allowing one to observe agent interactions and system dynamics in real time. The methodology section provides the most relevant code snippets illustrating how specific simulation aspects are programmed. The complete NetLogo code is provided in the Appendix for reference.

Data Collection & Data Cleaning

To ensure the accuracy of our agent-based model, we used data closely aligned with real-world conditions. While actual energy consumption data are incorporated into the simulation, assumptions were made to estimate solar energy production. This section outlines the data sources, data cleaning procedures, and assumptions applied. Data on households' energy consumption and potential PV production were collected for the German state of Baden-Württemberg (BW) for 2023. However, we are confident that the insights gained from this simulation are applicable beyond this geographic area, as the underlying decision-making mechanisms are not expected to be unique to this region.

Household Energy Consumption Data

To model the energy demand of individual households, we sourced data from the German Federal Network Agency (Bundesnetzagentur). Electricity consumption data for the control area "Transnet BW" were retrieved at an hourly resolution, representing the total electric energy demand for Baden-Württemberg. To isolate household energy consumption, the hourly values were multiplied by a coefficient of 0,26, reflecting that households account for 26% of the total grid load, while industrial users remain the largest energy consumers (Statistisches Landesamt Baden-Württemberg, 2023). To further break this data down to the household level, the total grid load values were converted from megawatt-hours (MWh) to kilowatt-hours (kWh) and then divided by the number of households in Baden-Württemberg. The number of households was estimated by dividing the state's population (11.300.000) by the average German household size of approximately two persons (McCarthy, 2019). Using this methodology, we determined the hourly energy consumption per household in Baden-Württemberg and integrated this data into the agent-based model.

Household Energy Production Data

It is assumed that energy produced from households comes exclusively from solar panels. To realistically calculate the potential energy production of households, data was retrieved from the Photovoltaic Geographical Information System (PVGIS) published by the European Commission. Stuttgart, the capital of Baden-Württemberg, was selected as the location for which radiation data was collected (hourly resolution). For determining the potential energy production from PV systems, the following assumptions were applied, motivated by real-world conditions and industry practice:

> *Assumption I*: Households have, on average, 20 solar panels installed
>
> *Assumption II*: Each solar panel has a power output of 440 Watts
>
> *Assumption III*: Optimal slope and azimuth are assumed for the installed panels with a 14% system loss (standard loss)

In reality, the number of panels a household decides to install depends on various factors, including the size, type, and inclination of the roof, as well as the potential shading of the panels from physical barriers like trees (Hoffmann, 2022). However, most households install between 15 and 30 panels based on cost-benefit considerations (Fonseca, 2024), justifying the assumption of 20 panels. This was further validated through an interview with industry expert Dr. Matthias Postina. Regarding panel power output, 440 Watts is considered the industry standard for new solar panels and is widely offered by leading manufacturers such as Jinko Solar and JA Solar. The system loss of 14% accounts for common inefficiencies, such as inverter losses and wiring resistance, and is a standard value used in PV simulations (PVGIS User Manual, 2024). Using these assumptions, hourly household energy production data was calculated for 2023. Since real radiation data was utilized, the values integrated into the ABM effectively capture the high fluctuations of PV energy production, ensuring the simulation reflects realistic dynamics.

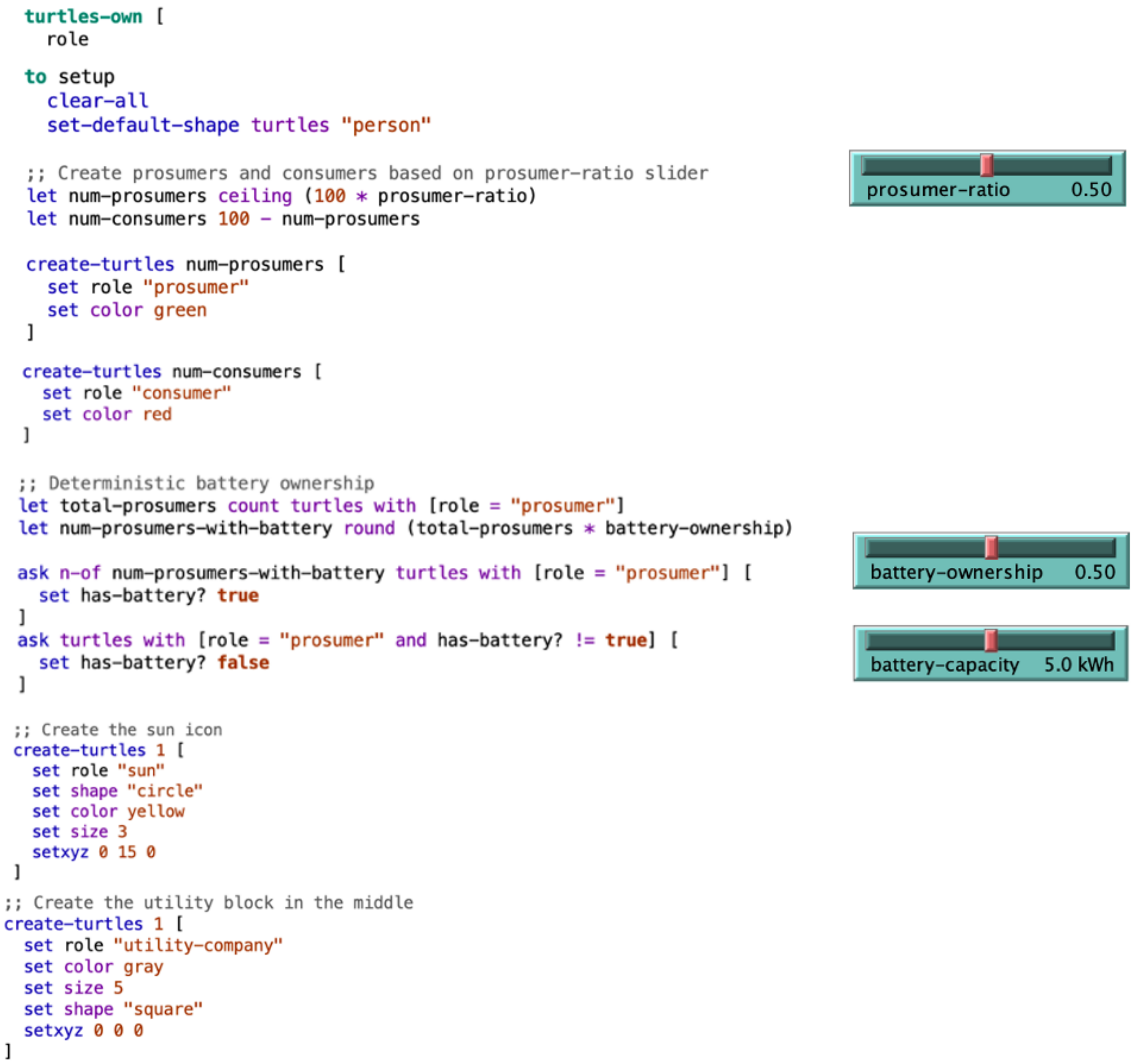
Introduction to the ABM

Simulation Setup

The ABM simulation designed for this thesis consists of 100 agents (referred to as "turtles"), each representing one household. Agents are assigned one of two roles: prosumer or consumer. Consumers only demand energy, whereas prosumers both produce and consume energy. Energy trading between agents allows consumers to meet part of their demand by purchasing surplus energy from prosumers. Whenever prosumers cannot fully meet the energy demand, the utility company steps in to supply the remaining energy.

In the simulation, consumers are visualized in red, while prosumers are represented in green as the default setting. Prosumers have the additional capability of storing surplus energy in their batteries. Surplus energy is defined as the energy remaining after a prosumer meets their immediate consumption needs. This surplus can either be traded with consumers or stored in their battery for later use, depending on the predefined rules governing agent behavior. The flexibility of the simulation is enhanced through interface sliders, which allow adjustments to key variables, such as the ratio of prosumers to consumers, battery storage capacity, and the proportion of prosumers with battery ownership. Other key elements of the energy trading system are represented as agents. The utility company acts as a backup energy provider to meet any unmet demand within the system, ensuring reliability. The sun serves as a visual representation in the simulation to symbolize the source of solar energy, reflecting the dependence of photovoltaic systems on solar radiation. Each agent's behavior and interactions are designed to mimic real-world dynamics, contributing to a realistic and insightful simulation of P2P energy trading.

All simulation components, including the roles and interactions of agents, are explained in greater detail in the following sections to provide a comprehensive understanding of the model's functionality and logic.



```
turtles-own [
  role

to setup
  clear-all
  set-default-shape turtles "person"

 ;; Create prosumers and consumers based on prosumer-ratio slider
 let num-prosumers ceiling (100 * prosumer-ratio)
 let num-consumers 100 - num-prosumers

 create-turtles num-prosumers [
   set role "prosumer"
   set color green
 ]

 create-turtles num-consumers [
   set role "consumer"
   set color red
 ]

 ;; Deterministic battery ownership
 let total-prosumers count turtles with [role = "prosumer"]
 let num-prosumers-with-battery round (total-prosumers * battery-ownership)

 ask n-of num-prosumers-with-battery turtles with [role = "prosumer"] [
   set has-battery? true
 ]
 ask turtles with [role = "prosumer" and has-battery? != true] [
   set has-battery? false
 ]

 ;; Create the sun icon
 create-turtles 1 [
   set role "sun"
   set shape "circle"
   set color yellow
   set size 3
   setxyz 0 15 0
 ]
;; Create the utility block in the middle
create-turtles 1 [
  set role "utility-company"
  set color gray
  set size 5
  set shape "square"
  setxyz 0 0 0
]
```

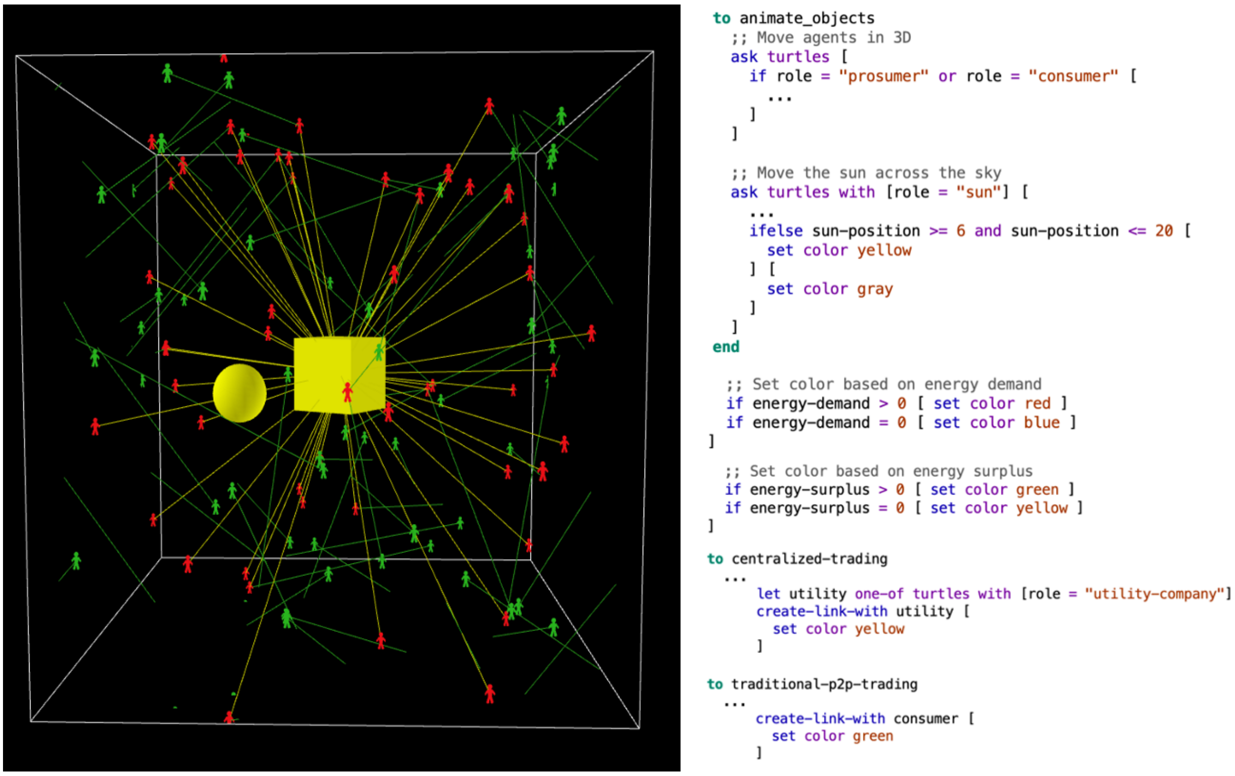
prosumer-ratio 0.50

battery-ownership 0.50

battery-capacity 5.0 kWh

Code Snippet 1: Setup

The 3D View illustrated below, serves as a visualization tool in our simulation, providing immediate visual feedback on the state of the system. By adjusting agent colors and visualizing trades through colored lines, we can observe the flow of energy, the engagement in P2P trading, and the reliance on the utility company.

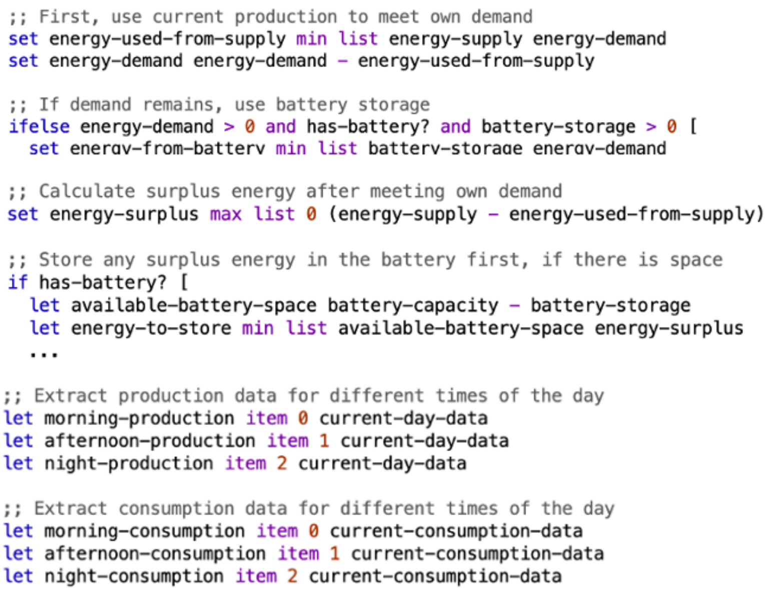


```
to animate_objects
  ;; Move agents in 3D
  ask turtles [
    if role = "prosumer" or role = "consumer" [
      ...
    ]
  ]

  ;; Move the sun across the sky
  ask turtles with [role = "sun"] [
    ...
    ifelse sun-position >= 6 and sun-position <= 20 [
      set color yellow
    ] [
      set color gray
    ]
  ]
end

  ;; Set color based on energy demand
  if energy-demand > 0 [ set color red ]
  if energy-demand = 0 [ set color blue ]
]

  ;; Set color based on energy surplus
  if energy-surplus > 0 [ set color green ]
  if energy-surplus = 0 [ set color yellow ]
]

to centralized-trading
  ...
      let utility one-of turtles with [role = "utility-company"]
      create-link-with utility [
        set color yellow
      ]

to traditional-p2p-trading
  ...
      create-link-with consumer [
        set color green
      ]
```

Code Snippet 2: 3D View

"Animate objects" makes agents move in the 3D environment, and the sun moves across the sky to represent the passage of time, changing color to indicate day and night cycles. This visual cue aligns with the agent’s energy production and consumption patterns, which depend on the time of day. The "set color" snippet ensures that each agent's color reflects its current energy state. The lines between agents represent energy flow through transactions, with the green line representing P2P energy trades between prosumers and consumers A yellow line indicates energy the utility supplies to agents with unmet energy demand.

Before exploring the three defined scenarios, we want to highlight further code snippets that outline some of the agents' predefined rules, decision-making processes, and data structure.



```
;; First, use current production to meet own demand
set energy-used-from-supply min list energy-supply energy-demand
set energy-demand energy-demand - energy-used-from-supply

;; If demand remains, use battery storage
ifelse energy-demand > 0 and has-battery? and battery-storage > 0 [
  set energy-from-battery min list battery-storage energy-demand

;; Calculate surplus energy after meeting own demand
set energy-surplus max list 0 (energy-supply - energy-used-from-supply)

;; Store any surplus energy in the battery first, if there is space
if has-battery? [
  let available-battery-space battery-capacity - battery-storage
  let energy-to-store min list available-battery-space energy-surplus
  ...

;; Extract production data for different times of the day
let morning-production item 0 current-day-data
let afternoon-production item 1 current-day-data
let night-production item 2 current-day-data

;; Extract consumption data for different times of the day
let morning-consumption item 0 current-consumption-data
let afternoon-consumption item 1 current-consumption-data
let night-consumption item 2 current-consumption-data
```

Code Snippet 3: Agent Decision-Making and Data Initialization

Prosumers prioritize meeting their energy demand, utilizing their battery storage as needed, and finally identifying any surplus energy. Their actions follow predefined rules, with trading behavior and battery logic varying across scenarios.

Model Documentation

For the documentation of an ABM, the Overview, Design Concepts, and Details (ODD) protocol, introduced by Grimm et al. (2006), is a widely accepted framework for standardizing the description of ABMs. ODD enhances the clarity and completeness of model documentation by providing a structured format that covers the model's purpose, variables, scales, concepts, and inputs. Akhatova et al. (2022) emphasize that a model's purpose must be clear, concise, and specific to help others understand why certain aspects of reality are included while others are omitted, a process they describe as "purposeful abstraction of reality." The ODD+D protocol extends the original framework by incorporating detailed explanations of human decision-making processes and implementation details. This extended version is particularly useful for documenting our simulation, as it captures the human decision-making aspects in detail, improving transparency and facilitating meaningful comparisons with other models. A detailed description of our model, using the ODD+D protocol, presented in text form and as a table, can be found in the Appendix.

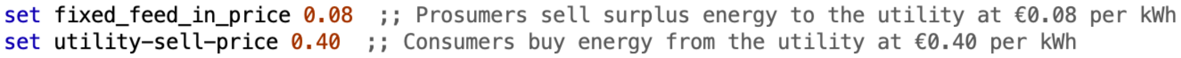
Scenarios

The following section presents a detailed explanation of the three scenarios analyzed in our simulation. Understanding the trading outcomes in these scenarios requires consideration of several key factors. P2P energy trading between prosumers and consumers occurs only when both parties benefit from the transaction. The following four price points influence their decisions:

> *Utility Price* (0,40€): Fixed price per kWh charged by the utility company
>
> *Feed-in Price* (0,08€): Fixed price per kWh that prosumers receive for feeding surplus energy into the grid
>
> *Prosumer Willingness to Sell* (AS2 only / slider set at 0,24€): Minimum price at which prosumers are willing to sell energy
>
> *P2P Price* (fixed at 0,24€ in AS1 and dynamic in AS2): Price at which prosumers sell energy to consumers in the P2P market. When dynamic pricing is enabled, the P2P Price varies based on total energy demand.

Base Scenario (BS): Central Energy Supply by the Utility Company

In the base scenario, all energy is supplied exclusively by the utility company. This setup models the common energy distribution system currently in place in Germany under an active feed-in-tariff scheme. In this scenario, prosumer households do not have the option to trade their produced energy directly with consumer households. Instead, they sell their surplus energy to the grid for a fixed feed-in rate of 0,08€/kWh in 2023 (Bundesnetzagentur, 2024). According to the Leipzig Energy Institute (2024), the average price per kilowatt-hour for end-users in Baden-Württemberg was 0,40€/kWh in 2023. Therefore, this value is used as the *Utility Price* in the simulation. Both *Feed-in Price* and *Utility Price* are initialized in the code as follows:



```
set fixed_feed_in_price 0.08  ;; Prosumers sell surplus energy to the utility at €0.08 per kWh
set utility-sell-price 0.40  ;; Consumers buy energy from the utility at €0.40 per kWh
```

Code Snippet 4: Prices

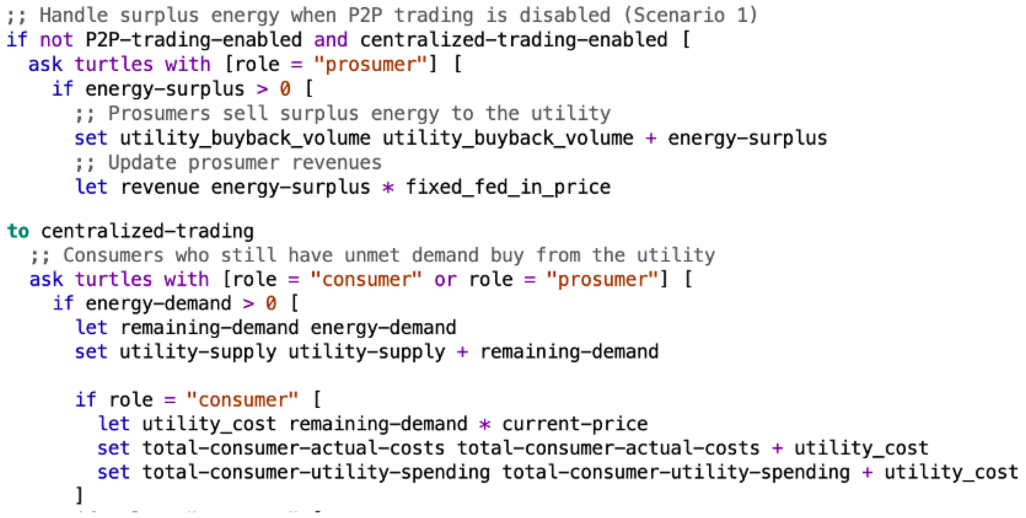
Prosumer Behavior

In the base scenario, prosumer behavior is straightforward as no P2P transactions occur between households. Prosumers produce energy using their PV systems and sell their surplus energy for the fixed *Feed-in Price* once their battery is fully charged. The following assumptions govern their actions in the simulation:

> *Assumption I:* Prosumer will always use their self-produced energy to first meet their own immediate demand
>
> *Assumption II:* When prosumers energy production exceeds their current demand, they will first store that surplus energy in their battery
>
> *Assumption III:* Once prosumer's batteries are fully charged, any remaining surplus energy is sent to the grid at the fixed *Feed-in Price*

Consumer Behavior

Consumers rely entirely on the utility company for their energy supply. They purchase all their energy at the *Utility Price* of 0,40€. The following code snippet captures the behavior of prosumers under the feed-in-tariff scheme and consumers purchasing energy from the utility at the fixed *Utility Price*:



```
;; Handle surplus energy when P2P trading is disabled (Scenario 1)
if not P2P-trading-enabled and centralized-trading-enabled [
  ask turtles with [role = "prosumer"] [
    if energy-surplus > 0 [
      ;; Prosumers sell surplus energy to the utility
      set utility_buyback_volume utility_buyback_volume + energy-surplus
      ;; Update prosumer revenues
      let revenue energy-surplus * fixed_fed_in_price

to centralized-trading
  ;; Consumers who still have unmet demand buy from the utility
  ask turtles with [role = "consumer" or role = "prosumer"] [
    if energy-demand > 0 [
      let remaining-demand energy-demand
      set utility-supply utility-supply + remaining-demand

      if role = "consumer" [
        let utility_cost remaining-demand * current-price
        set total-consumer-actual-costs total-consumer-actual-costs + utility_cost
        set total-consumer-utility-spending total-consumer-utility-spending + utility_cost
      ]
```

Code Snippet 5: Base Scenario

Alternative Scenario 1 (AS1): P2P Energy Trading enabled, without Dynamic Pricing

In AS1, prosumers can trade their surplus energy directly with consumers at a fixed *P2P Price* of 0,24€/kWh or sell it to the grid at the *Feed-in Price* of 0,08€/kWh. The fixed *P2P Price* is set midway between the *Feed-in Price* and the *Utility Price*, ensuring that both prosumers and consumers benefit equally from trading. The goal of this scenario is to investigate how agents behave when given the opportunity to trade P2P. The number of conducted P2P trades, financial benefits, and energy autonomy are of special interest in this regard.

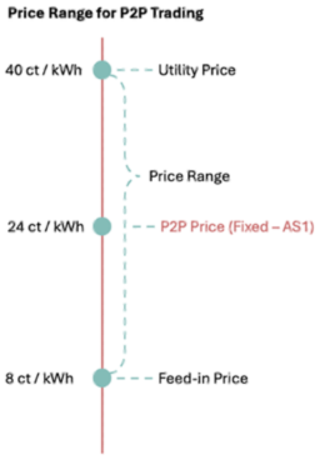


Figure 1: Price Range P2P Trading

Prosumer Behavior

Once a prosumer's battery is fully charged, they trade their surplus energy directly with consumers at the *P2P Price* since it exceeds the *Feed-in Price* they would receive, sending it into the grid. The following assumptions guide prosumers' behavior:

> *Assumption I:* Prosumer will always use their self-produced energy to first meet their own immediate demand
>
> *Assumption II:* When prosumers energy production exceeds their current demand, they will first store that surplus energy in their battery

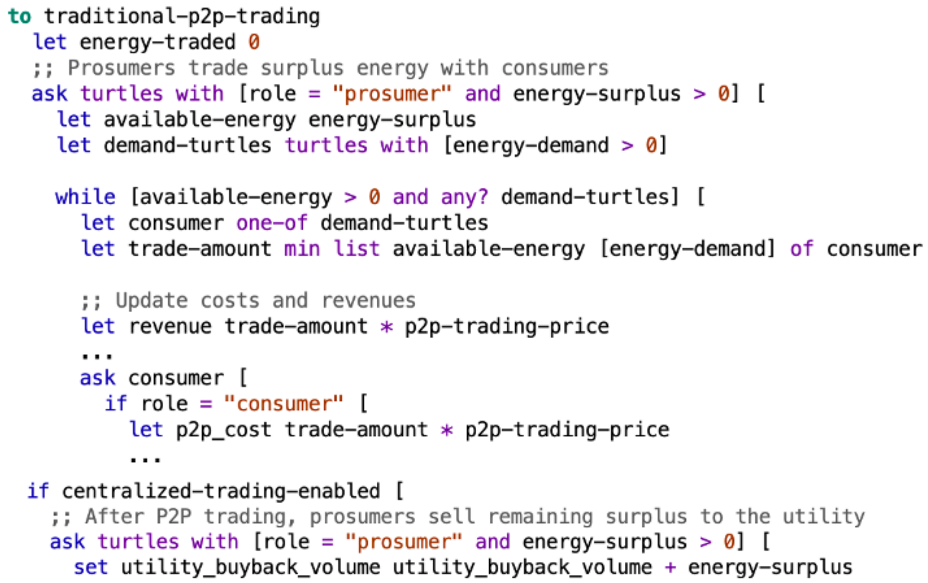
> *Assumption III:* Once prosumer's batteries are fully charged, they decide to trade surplus energy with consumers at the fixed *P2P Price* or send it to the grid for the *Feed-in Price* if there is no demand from consumers.

Consumer Behavior

With P2P trading enabled consumers have the option to meet their energy demand by purchasing energy directly from prosumers. Their decision to buy from prosumers or the utility is driven solely by price and the associated cost savings. Consumers prefer to buy energy from prosumers when the *P2P Price* is below the *Utility Price*. The following assumption governs consumers' behavior:

> *Assumption I*: Consumers always opt for the lowest available price, regardless of whether the energy comes from peers or the utility company

P2P Trading, where prosumers and consumers engage in transactions based on predefined pricing rules, is initialized in the code as follows:



```
to traditional-p2p-trading
  let energy-traded 0
  ;; Prosumers trade surplus energy with consumers
  ask turtles with [role = "prosumer" and energy-surplus > 0] [
    let available-energy energy-surplus
    let demand-turtles turtles with [energy-demand > 0]

    while [available-energy > 0 and any? demand-turtles] [
      let consumer one-of demand-turtles
      let trade-amount min list available-energy [energy-demand] of consumer

      ;; Update costs and revenues
      let revenue trade-amount * p2p-trading-price
      ...
      ask consumer [
        if role = "consumer" [
          let p2p_cost trade-amount * p2p-trading-price
          ...
  if centralized-trading-enabled [
    ;; After P2P trading, prosumers sell remaining surplus to the utility
    ask turtles with [role = "prosumer" and energy-surplus > 0] [
      set utility_buyback_volume utility_buyback_volume + energy-surplus
```

Code Snippet 6: P2P Trading (AS1)

Alternative Scenario 2 (AS2): Enabled P2P Energy Trading under Dynamic Pricing

In AS2, dynamic pricing is introduced to create a more realistic representation of energy trading, with prices fluctuating based on demand levels. In this scenario, prosumers may also trade stored energy, adding complexity to their trading decisions. Prosumers have a psychological cut-off value, referred to as *Willingness to Sell*, which is set to 0,24€ via a slider. This value represents the threshold at which prosumers are willing to participate in P2P trading, balancing financial incentives for prosumers with affordability for consumers, as it lies between the *Feed-in Price* (0,08€) and the *Utility Price* (0,40€). While any price above the *Feed-in Price* is technically profitable, prosumers consider the uncertainty of future energy production and the potential need to buy back energy from the utility at a higher price. These dynamics and the detailed battery storage logic are elaborated in the section below. The dynamic *P2P Price* fluctuates within this range, rising when demand is high and falling when it is low, ensuring a fair and demand-responsive market (Figure 2). The two examples of *P2P Prices* shown in orange in the figure are illustrative values chosen to clarify the concept.

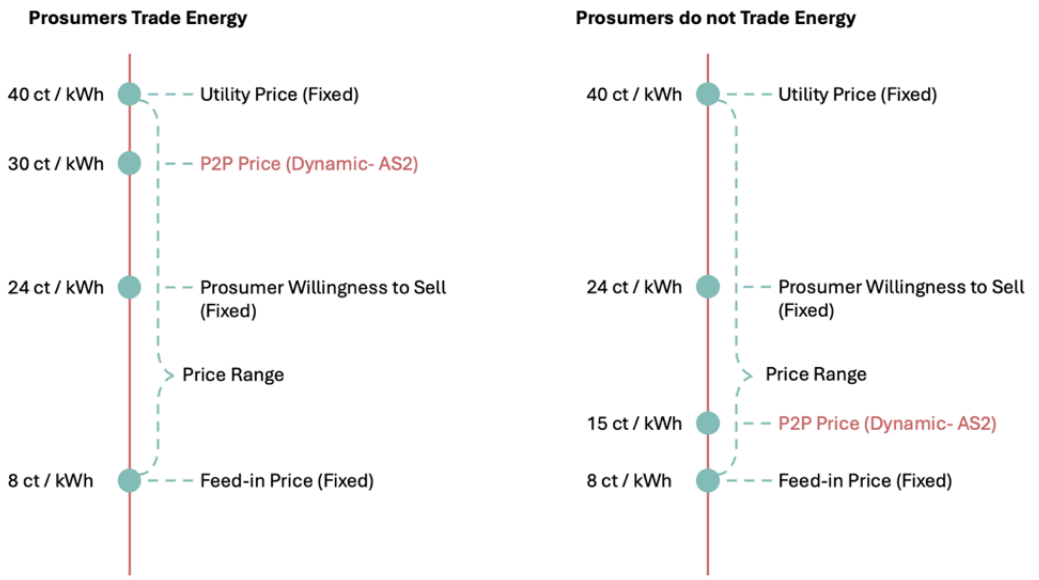
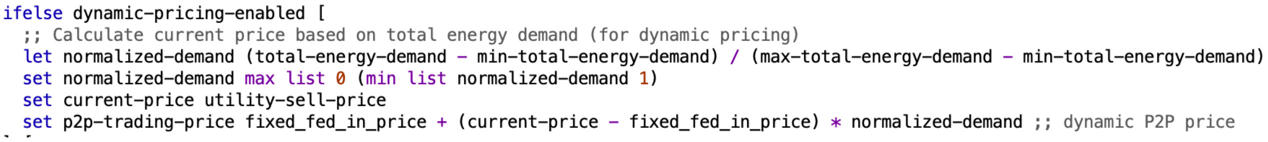


Figure 2: Comparison of Pricing Dynamics: P2P Energy Trading vs. No Trading

The logic was implemented in the code as follows:



```
ifelse dynamic-pricing-enabled [
  ;; Calculate current price based on total energy demand (for dynamic pricing)
  let normalized-demand (total-energy-demand - min-total-energy-demand) / (max-total-energy-demand - min-total-energy-demand)
  set normalized-demand max list 0 (min list normalized-demand 1)
  set current-price utility-sell-price
  set p2p-trading-price fixed_fed_in_price + (current-price - fixed_fed_in_price) * normalized-demand ;; dynamic P2P price
```

Code Snippet 7: Dynamic P2P Price

This pricing logic adjusts the P2P Price dynamically based on total energy demand. The normalized-demand variable scales demand between 0 and 1, and the P2P Price is calculated as a weighted value within the range, ensuring economic viability for consumers and prosumers.

Prosumer Behavior

In this scenario, trading occurs when the dynamic *P2P Price* is equal to or exceeds the *Willingness to Sell*, incentivizing prosumers to engage in P2P trading. Unlike in AS1, prosumers in AS2 may also trade energy stored in their batteries, provided their storage level is at least 50%. This introduces additional flexibility and opportunities for trading beyond simply selling surplus energy when batteries are full. The logic governing battery storage and trading decisions will be analyzed further in the next section. The following assumptions guide prosumers' behavior in AS2:

> *Assumption I:* Prosumer will always use their self-produced energy to first meet their own immediate demand
>
> *Assumption II:* When prosumers energy production exceeds their current demand, they will start storing surplus energy in their batteries
>
> *Assumption III:* Once prosumer's batteries are fully charged, they decide to trade surplus energy with consumers if the dynamic *P2P Price* is greater or equal to their *Willingness to Sell*
>
> *Assumption IV:* Prosumers may also sell stored energy, even if the batteries are not fully charged, provided that the remaining charge stays above the 50% safety threshold to ensure future energy needs can be met

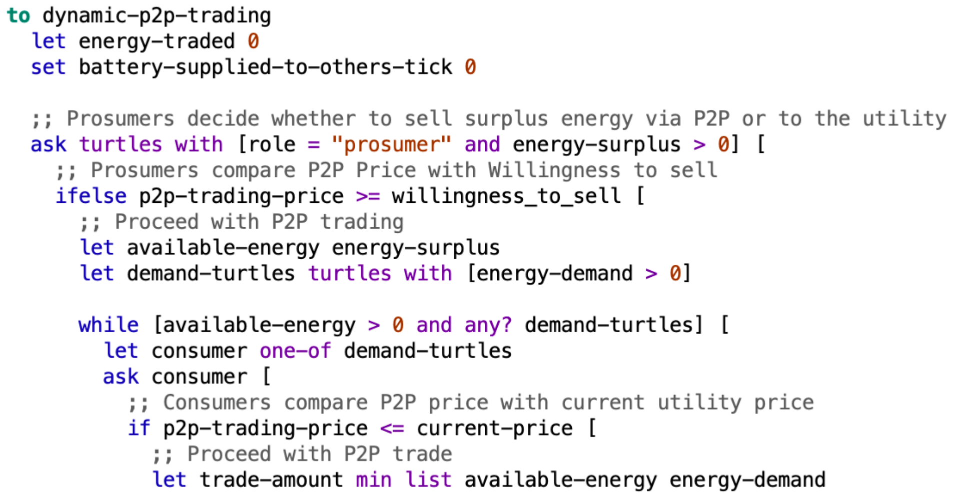
Consumer Behavior

Consumers compare the dynamic *P2P Price* with the *Utility Price* and always choose the cheapest option. Any *P2P Price* below the *Utility Price* makes economic sense for consumers. The following assumption governs consumers' behavior:

> *Assumption I:* Consumers always opt for the lowest available price, regardless of whether the energy comes from peers or the utility company

By following this logic, consumers achieve cost savings by purchasing energy at rates below the *Utility Price*, while prosumers generate additional revenue by selling energy at prices above the *Feed-in Price*. This mutual benefit incentivizes both parties to participate in P2P energy trading, maximizing the potential of this localized and decentralized market.

Dynamic Pricing is implemented in the code as follows:

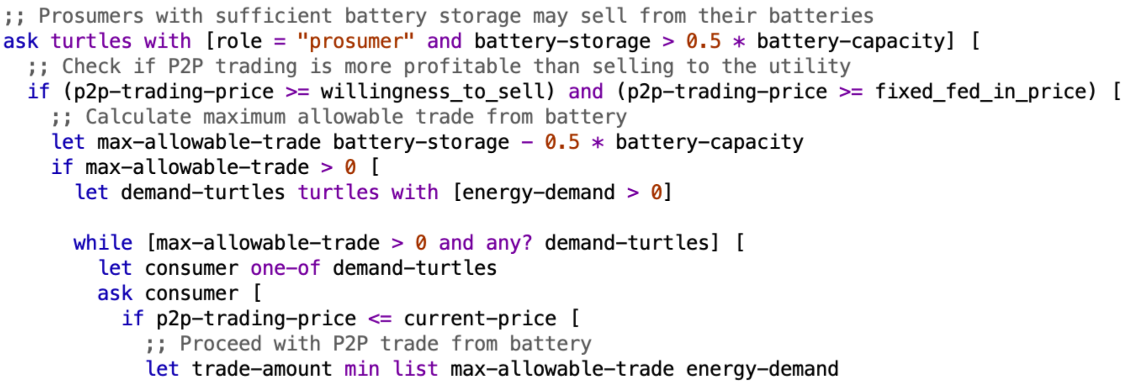


```
to dynamic-p2p-trading
  let energy-traded 0
  set battery-supplied-to-others-tick 0

  ;; Prosumers decide whether to sell surplus energy via P2P or to the utility
  ask turtles with [role = "prosumer" and energy-surplus > 0] [
    ;; Prosumers compare P2P Price with Willingness to sell
    ifelse p2p-trading-price >= willingness_to_sell [
      ;; Proceed with P2P trading
      let available-energy energy-surplus
      let demand-turtles turtles with [energy-demand > 0]

      while [available-energy > 0 and any? demand-turtles] [
        let consumer one-of demand-turtles
        ask consumer [
          ;; Consumers compare P2P price with current utility price
          if p2p-trading-price <= current-price [
            ;; Proceed with P2P trade
            let trade-amount min list available-energy energy-demand
```

Code Snippet 8: Dynamic Pricing

Prosumer Battery Logic and Energy Storage Decision

The battery logic in our simulation varies depending on the scenario. In AS1, prosumers fully charge their batteries before trading any surplus energy. In AS2, prosumers have more flexibility, allowing them to trade energy even if their batteries are not fully charged. This enables them to capitalize on immediate financial opportunities when dynamic *P2P Prices* are favorable. However, this approach introduces the risk that prosumers may trade energy they will later need to meet their own consumption demands. To mitigate this risk, a threshold is introduced, requiring prosumers to maintain at least a 50% battery charge. This safeguard ensures that prosumers can still meet most of their own energy needs independently. Since buying energy from the utility is always a loss-making option for prosumers, given that the *Utility Price* is higher than the dynamic *P2P Price*, they are incentivized to manage their stored energy strategically. This battery logic balances the financial gains from trading energy with the uncertainty of future energy production. The logic was implemented in the code as follows:



```
;; Prosumers with sufficient battery storage may sell from their batteries
ask turtles with [role = "prosumer" and battery-storage > 0.5 * battery-capacity] [
  ;; Check if P2P trading is more profitable than selling to the utility
  if (p2p-trading-price >= willingness_to_sell) and (p2p-trading-price >= fixed_fed_in_price) [
    ;; Calculate maximum allowable trade from battery
    let max-allowable-trade battery-storage - 0.5 * battery-capacity
    if max-allowable-trade > 0 [
      let demand-turtles turtles with [energy-demand > 0]

      while [max-allowable-trade > 0 and any? demand-turtles] [
        let consumer one-of demand-turtles
        ask consumer [
          if p2p-trading-price <= current-price [
            ;; Proceed with P2P trade from battery
            let trade-amount min list max-allowable-trade energy-demand
```

Code Snippet 9: Battery Logic

In a real-world scenario, prosumers could use forecasting techniques to predict future energy needs based on expected production and consumption patterns, enabling more precise trading decisions. However, this simulation employs a 50% buffer as a practical solution to account for uncertainties without introducing the complexity of forecasting models. This approach ensures a reliable energy reserve, safeguarding against potential shortages if production falls short of expectations. By simplifying the decision-making process, the buffer allows the ABM to focus on capturing the core dynamics of P2P energy trading without overcomplicating the simulation. Additionally, this method provides a solid baseline that can be built upon with more advanced techniques in future iterations, allowing for added complexity while maintaining the model's clarity and accessibility.

References

Akhatova, Ardak, Lukas Kranzl, Fabian Schipfer, and Charitha Buddhika Heendeniya. “Agent-Based Modelling of Urban District Energy System Decarbonisation—A Systematic Literature Review.” *Energies* 15, no. 2 (January 13, 2022): 554. https://doi.org/10.3390/en15020554.

Alam, Muhammad Raisul, Marc St-Hilaire, and Thomas Kunz. “Peer-to-peer energy trading among smart homes.” *Applied Energy* 238 (January 30, 2019): 1434–43. https://doi.org/10.1016/j.apenergy.2019.01.091.

Andoni, Merlinda, Valentin Robu, David Flynn, Simone Abram, Dale Geach, David Jenkins, Peter McCallum, and Andrew Peacock. “Blockchain technology in the energy sector: A systematic review of challenges and opportunities.” *Renewable and Sustainable Energy Reviews* 100 (November 3, 2018): 143–74. https://doi.org/10.1016/j.rser.2018.10.014.

“Bundesnetzagentur - EEG-Förderung und -Fördersätze,” n.d. https://www.bundesnetzagentur.de/DE/Fachthemen/ElektrizitaetundGas/ErneuerbareEnergien/EEG_Foerderung/start.html.

Cali, Umit, and Austin Fifield. “Towards the decentralized revolution in energy systems using blockchain technology.” *International Journal of Smart Grid and Clean Energy*, January 1, 2019, 245–56. https://doi.org/10.12720/sgce.8.3.245-256.

Dr. Matthias Postina. Interview by Fabian Simon and Tim Rempe. October 15, 2024.

“Energiebericht kompakt 2023 - Statistisches Landesamt Baden-Württemberg,” June 21, 2023. https://www.statistik-bw.de/Presse/Pressemitteilungen/2023140.

Erfurt, Ilka, and Matthias Reichmuth. “Preisbericht für den Energiemarkt in Baden-Württemberg 2023.” Edited by Alexander Schiffler, Marcel Ebert, Lisa Horbach, Robert Kießling, Matthias Reichmuth, and Andreas Weber. *ENDBERICHT*, July 5,

2024. https://www.baden-wuerttemberg.de/fileadmin/redaktion/m-um/intern/Dateien/Dokumente/5_Energie/Versorgungssicherheit/Energiepreise/240705-Energiepreisbericht-2023.

Evens, Maarten, Patricia Ercoli, and Alessia Arteconi. "Blockchain-Enabled Microgrids: Toward Peer-to-Peer Energy Trading and Flexible Demand Management." *Energies* 16, no. 18 (September 21, 2023): 6741. https://doi.org/10.3390/en16186741.

Fonseca, Stefano. "Wie viel Photovoltaik und Solarmodule brauche ich in 2024?" homeandsmart.de, January 24, 2024. https://www.homeandsmart.de/solarmodule-anzahl-ermitteln.

German Solar Association (BSW-Solar). "Statistical data on the German Solar Battery Storage & E-Mobility Market." Statistical data. Berlin, 2024. https://www.solarwirtschaft.de/datawall/uploads/2020/08/bsw_factsheet_solar_battery_storage_emob_eng.pdf.

"Germany targets variable customer power prices by 2025 | Montel News - English," n.d. https://montelnews.com/news/1404068/germany-targets-variable-customer-power-prices-by-2025.

Grimm, Volker, Uta Berger, Finn Bastiansen, Sigrunn Eliassen, Vincent Ginot, Jarl Giske, John Goss-Custard, et al. "A standard protocol for describing individual-based and agent-based models." *Ecological Modelling* 198, no. 1–2 (June 15, 2006): 115–26. https://doi.org/10.1016/j.ecolmodel.2006.04.023.

Guimaraes, Diogo V., Matthew B Gough, Sergio F. Santos, Ines F.G. Reis, Juan M. Home-Ortiz, and Joao P. S. Catalao. "Agent-Based Modeling of Peer-to-Peer Energy Trading in a Smart Grid Environment." *2022 IEEE International Conference on Environment and Electrical Engineering and 2022 IEEE Industrial and Commercial Power Systems*

Europe (EEEIC / I&Amp;CPS Europe), September 7, 2021, 1–6. https://doi.org/10.1109/eeeic/icpseurope51590.2021.9584767.

Hoffmann, Nina. "Wie viele Solarmodule brauche ich?" *Forbes Advisor Deutschland*, November 1, 2022. https://www.forbes.com/advisor/de/energie/solarenergie/wie-viele-solarmodule/.

"JRC Photovoltaic Geographical Information System (PVGIS) - European Commission," January 11, 2016. https://re.jrc.ec.europa.eu/pvg_tools/en/#api_5.3.

Liu, Nian, Xinghuo Yu, Cheng Wang, Chaojie Li, Li Ma, and Jinyong Lei. "Energy-Sharing Model With Price-Based Demand Response for Microgrids of Peer-to-Peer Prosumers." *IEEE Transactions on Power Systems* 32, no. 5 (January 9, 2017): 3569–83. https://doi.org/10.1109/tpwrs.2017.2649558.

Long, Chao, Yue Zhou, and Jianzhong Wu. "A game theoretic approach for peer to peer energy trading." *Energy Procedia* 159 (February 1, 2019): 454–59. https://doi.org/10.1016/j.egypro.2018.12.075.

McCarthy, Niall. "German households are getting smaller." *Statista Daily Data*, July 26, 2019. https://www.statista.com/chart/18820/average-number-of-household-members-in-german-federal-states/.

Mengelkamp, Esther, Johannes Gärttner, Kerstin Rock, Scott Kessler, Lawrence Orsini, and Christof Weinhardt. "Designing microgrid energy markets." *Applied Energy* 210 (June 23, 2017): 870–80. https://doi.org/10.1016/j.apenergy.2017.06.054.

Monroe, Jacob G., Paula Hansen, Matthew Sorell, and Emily Zechman Berglund. "Agent-Based Model of a Blockchain Enabled Peer-to-Peer Energy Market: Application for a Neighborhood Trial in Perth, Australia." *Smart Cities* 3, no. 3 (September 19, 2020): 1072–99. https://doi.org/10.3390/smartcities3030053.

Mylrea, Michael, and Sri Nikhil Gupta Gourisetti. “Blockchain for smart grid resilience: Exchanging distributed energy at speed, scale and security.” *IEEE*, September 1, 2017. https://doi.org/10.1109/rweek.2017.8088642.

Nguyen, Su, Wei Peng, Peter Sokolowski, Damminda Alahakoon, and Xinghuo Yu. “Optimizing rooftop photovoltaic distributed generation with battery storage for peer-to-peer energy trading.” *Applied Energy* 228 (September 3, 2018): 2567–80. https://doi.org/10.1016/j.apenergy.2018.07.042.

Powerledger. “Home - Powerledger,” November 19, 2024. https://powerledger.io/.

Saltouh, Tarek M. E. Abou, Abd El-Shafy A. Nafeh, Adel a. Abou El-Ela, Faten H. Fahmy, and Sherif K. Nawar. “Control strategy for seamless transition between grid-connected and islanding modes in microgrid-based PV inverters.” *Energy Systems* 14, no. 4 (August 25, 2022): 1135–62. https://doi.org/10.1007/s12667-022-00528-1.

SMARD. “SMARD | SMARD - Market data,” November 27, 2024. https://www.smard.de/en.

Sonnen. “sonnenCommunity | sonnen,” n.d. https://sonnen.de/sonnencommunity/.

Terralayr. “energy flexibility. on demand,” n.d. https://www.trlyr.com/.

Tushar, Wayes, Tapan Kumar Saha, Chau Yuen, Thomas Morstyn, Malcolm D. McCulloch, H. Vincent Poor, and Kristin L. Wood. “A motivational game-theoretic approach for peer-to-peer energy trading in the smart grid.” *Applied Energy* 243 (March 30, 2019): 10–20. https://doi.org/10.1016/j.apenergy.2019.03.111.

Umar, Abdullah, Deepak Kumar, and Tirthadip Ghose. “Decentralized energy trading in microgrids: a blockchain-integrated model for efficient power flow with social welfare optimization.” *Electrical Engineering*, August 21, 2024. https://doi.org/10.1007/s00202-024-02635-x.

Wongthongtham, Pornpit, Daniel Marrable, Bilal Abu-Salih, Xin Liu, and Greg Morrison. “Blockchain-enabled Peer-to-Peer energy trading.” *Computers & Electrical*

Engineering 94 (July 10, 2021): 107299. https://doi.org/10.1016/j.compeleceng.2021.107299.

Wu, Shunxiang, Fangfei Zhang, and Danlin Li. “User-Centric Peer-to-Peer Energy Trading Mechanisms for Residential Microgrids.” *IEEE*, October 1, 2018, 1–6. https://doi.org/10.1109/ei2.2018.8582548.

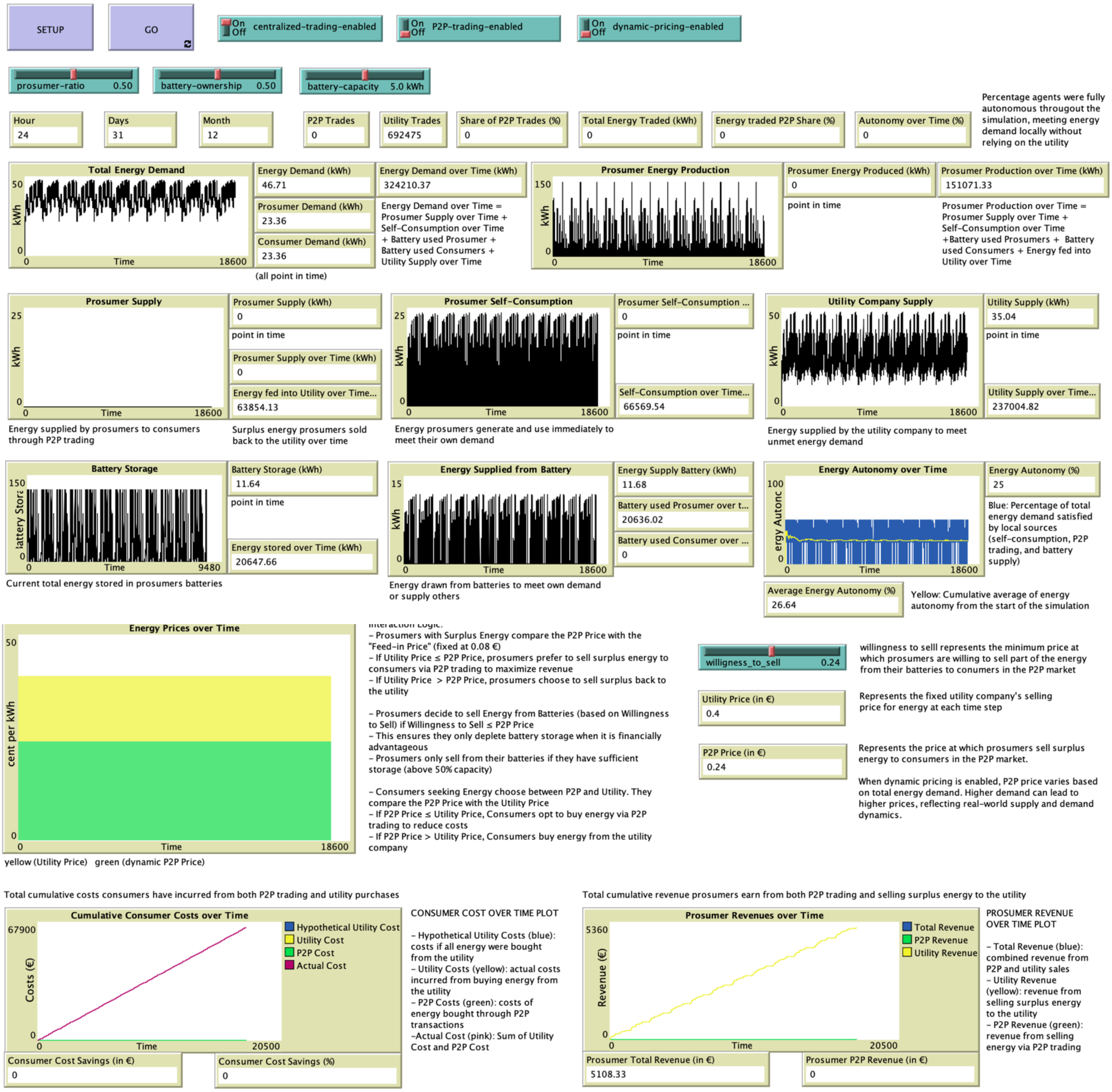
Yao, Ruiqiu, Yukun Hu, and Liz Varga. “Applications of Agent-Based Methods in Multi-Energy Systems—A Systematic Literature Review.” *Energies* 16, no. 5 (March 4, 2023): 2456. https://doi.org/10.3390/en16052456.

Zhang, Zhenyuan, Haoyue Tang, Qi Huang, and Wei-Jen Lee. “Two-Stages Bidding Strategies for Residential Microgrids Based Peer-to-Peer Energy Trading.” *IEEE*, May 1, 2019. https://doi.org/10.1109/icps.2019.8733335.

Appendix

Net Logo Energy Trading Simulation Access Link:
https://drive.google.com/file/d/1_GHZmR1ZMRBY901r8yfGyL6d-5FcHKWw/view?usp=sharing

NetLogo Simulation Interface



On Off centralized-trading-enabled

On Off P2P-trading-enabled

On Off dynamic-pricing-enabled

prosumer-ratio 0.50

battery-ownership 0.50

battery-capacity 5.0 kWh

Hour: 24

Days: 31

Month: 12

P2P Trades: 0

Utility Trades: 692475

Share of P2P Trades (%): 0

Total Energy Traded (kWh): 0

Energy traded P2P Share (%): 0

Autonomy over Time (%): 0

Percentage agents were fully autonomous througout the simulation, meeting energy demand locally without relying on the utility

Energy Demand (kWh): 46.71

Prosumer Demand (kWh): 23.36

Consumer Demand (kWh): 23.36

(all point in time)

Energy Demand over Time (kWh): 324210.37

Energy Demand over Time = Prosumer Supply over Time + Self-Consumption over Time + Battery used Prosumer + Battery used Consumers + Utility Supply over Time

Prosumer Energy Produced (kWh): 0

point in time

Prosumer Production over Time (kWh): 151071.33

Prosumer Production over Time = Prosumer Supply over Time + Self-Consumption over Time +Battery used Prosumers + Battery used Consumers + Energy fed into Utility over Time

Prosumer Supply (kWh): 0

point in time

Prosumer Supply over Time (kWh): 0

Energy fed into Utility over Time...: 63854.13

Energy supplied by prosumers to consumers through P2P trading

Surplus energy prosumers sold back to the utility over time

Prosumer Self-Consumption ...: 0

point in time

Self-Consumption over Time...: 66569.54

Energy prosumers generate and use immediately to meet their own demand

Utility Supply (kWh): 35.04

point in time

Utility Supply over Time...: 237004.82

Energy supplied by the utility company to meet unmet energy demand

Battery Storage (kWh): 11.64

point in time

Energy stored over Time (kWh): 20647.66

Current total energy stored in prosumers batteries

Energy Supply Battery (kWh): 11.68

Battery used Prosumer over t...: 20636.02

Battery used Consumer over ...: 0

Energy drawn from batteries to meet own demand or supply others

Energy Autonomy (%): 25

Blue: Percentage of total energy demand satisfied by local sources (self-consumption, P2P trading, and battery supply)

Average Energy Autonomy (%): 26.64

Yellow: Cumulative average of energy autonomy from the start of the simulation

yellow (Utility Price) green (dynamic P2P Price)

Interaction Logic:
- Prosumers with Surplus Energy compare the P2P Price with the "Feed-in Price" (fixed at 0.08 €)
- If Utility Price ≤ P2P Price, prosumers prefer to sell surplus energy to consumers via P2P trading to maximize revenue
- If Utility Price > P2P Price, prosumers choose to sell surplus back to the utility

- Prosumers decide to sell Energy from Batteries (based on Willingness to Sell) if Willingness to Sell ≤ P2P Price
- This ensures they only deplete battery storage when it is financially advantageous
- Prosumers only sell from their batteries if they have sufficient storage (above 50% capacity)

- Consumers seeking Energy choose between P2P and Utility. They compare the P2P Price with the Utility Price
- If P2P Price ≤ Utility Price, Consumers opt to buy energy via P2P trading to reduce costs
- If P2P Price > Utility Price, Consumers buy energy from the utility company

willingness_to_sell 0.24

willingness to selll represents the minimum price at which prosumers are willing to sell part of the energy from their batteries to conumers in the P2P market

Utility Price (in €): 0.4

Represents the fixed utility company's selling price for energy at each time step

P2P Price (in €): 0.24

Represents the price at which prosumers sell surplus energy to consumers in the P2P market.

When dynamic pricing is enabled, P2P price varies based on total energy demand. Higher demand can lead to higher prices, reflecting real-world supply and demand dynamics.

Total cumulative costs consumers have incurred from both P2P trading and utility purchases

🟦 Hypothetical Utility Cost
🟨 Utility Cost
🟩 P2P Cost
🟥 Actual Cost

Consumer Cost Savings (in €): 0

Consumer Cost Savings (%): 0

CONSUMER COST OVER TIME PLOT

- Hypothetical Utility Costs (blue): costs if all energy were bought from the utility
- Utility Costs (yellow): actual costs incurred from buying energy from the utility
- P2P Costs (green): costs of energy bought through P2P transactions
-Actual Cost (pink): Sum of Utility Cost and P2P Cost

Total cumulative revenue prosumers earn from both P2P trading and selling surplus energy to the utility

🟦 Total Revenue
🟩 P2P Revenue
🟨 Utility Revenue

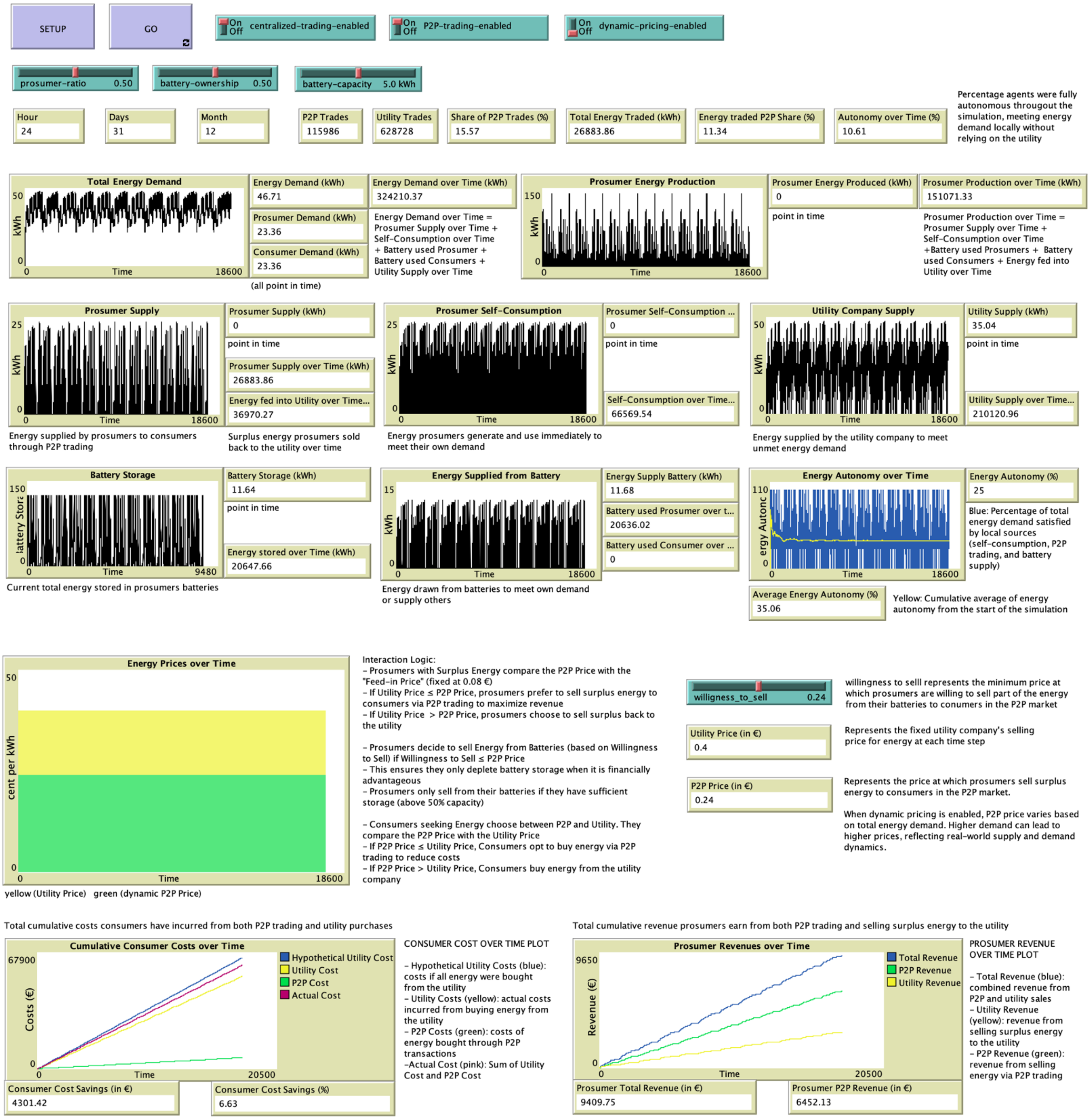
Prosumer Total Revenue (in €): 5108.33

Prosumer P2P Revenue (in €): 0

PROSUMER REVENUE OVER TIME PLOT

- Total Revenue (blue): combined revenue from P2P and utility sales
- Utility Revenue (yellow): revenue from selling surplus energy to the utility
- P2P Revenue (green): revenue from selling energy via P2P trading

Figure 3: Simulation Interface - Base Scenario



prosumer-ratio 0.50

battery-ownership 0.50

battery-capacity 5.0 kWh

Hour 24

Days 31

Month 12

P2P Trades 115986

Utility Trades 628728

Share of P2P Trades (%) 15.57

Total Energy Traded (kWh) 26883.86

Energy traded P2P Share (%) 11.34

Autonomy over Time (%) 10.61

Percentage agents were fully autonomous througout the simulation, meeting energy demand locally without relying on the utility

Energy Demand (kWh) 46.71

Prosumer Demand (kWh) 23.36

Consumer Demand (kWh) 23.36

(all point in time)

Energy Demand over Time (kWh) 324210.37

Energy Demand over Time = Prosumer Supply over Time + Self-Consumption over Time + Battery used Prosumer + Battery used Consumers + Utility Supply over Time

Prosumer Energy Produced (kWh) 0

point in time

Prosumer Production over Time (kWh) 151071.33

Prosumer Production over Time = Prosumer Supply over Time + Self-Consumption over Time +Battery used Prosumers + Battery used Consumers + Energy fed into Utility over Time

Energy supplied by prosumers to consumers through P2P trading

Prosumer Supply (kWh) 0

point in time

Prosumer Supply over Time (kWh) 26883.86

Energy fed into Utility over Time... 36970.27

Surplus energy prosumers sold back to the utility over time

Energy prosumers generate and use immediately to meet their own demand

Prosumer Self-Consumption ... 0

point in time

Self-Consumption over Time... 66569.54

Energy supplied by the utility company to meet unmet energy demand

Utility Supply (kWh) 35.04

point in time

Utility Supply over Time... 210120.96

Current total energy stored in prosumers batteries

Battery Storage (kWh) 11.64

point in time

Energy stored over Time (kWh) 20647.66

Energy drawn from batteries to meet own demand or supply others

Energy Supply Battery (kWh) 11.68

Battery used Prosumer over t... 20636.02

Battery used Consumer over ... 0

Energy Autonomy (%) 25

Blue: Percentage of total energy demand satisfied by local sources (self-consumption, P2P trading, and battery supply)

Average Energy Autonomy (%) 35.06

Yellow: Cumulative average of energy autonomy from the start of the simulation

yellow (Utility Price) green (dynamic P2P Price)

Interaction Logic:
- Prosumers with Surplus Energy compare the P2P Price with the "Feed-in Price" (fixed at 0.08 €)
- If Utility Price ≤ P2P Price, prosumers prefer to sell surplus energy to consumers via P2P trading to maximize revenue
- If Utility Price > P2P Price, prosumers choose to sell surplus back to the utility

- Prosumers decide to sell Energy from Batteries (based on Willingness to Sell) if Willingness to Sell ≤ P2P Price
- This ensures they only deplete battery storage when it is financially advantageous
- Prosumers only sell from their batteries if they have sufficient storage (above 50% capacity)

- Consumers seeking Energy choose between P2P and Utility. They compare the P2P Price with the Utility Price
- If P2P Price ≤ Utility Price, Consumers opt to buy energy via P2P trading to reduce costs
- If P2P Price > Utility Price, Consumers buy energy from the utility company

willingess_to_sell 0.24

willingness to selll represents the minimum price at which prosumers are willing to sell part of the energy from their batteries to conumers in the P2P market

Utility Price (in €) 0.4

Represents the fixed utility company's selling price for energy at each time step

P2P Price (in €) 0.24

Represents the price at which prosumers sell surplus energy to consumers in the P2P market.

When dynamic pricing is enabled, P2P price varies based on total energy demand. Higher demand can lead to higher prices, reflecting real-world supply and demand dynamics.

Total cumulative costs consumers have incurred from both P2P trading and utility purchases

Consumer Cost Savings (in €) 4301.42

Consumer Cost Savings (%) 6.63

CONSUMER COST OVER TIME PLOT

- Hypothetical Utility Costs (blue): costs if all energy were bought from the utility
- Utility Costs (yellow): actual costs incurred from buying energy from the utility
- P2P Costs (green): costs of energy bought through P2P transactions
-Actual Cost (pink): Sum of Utility Cost and P2P Cost

Total cumulative revenue prosumers earn from both P2P trading and selling surplus energy to the utility

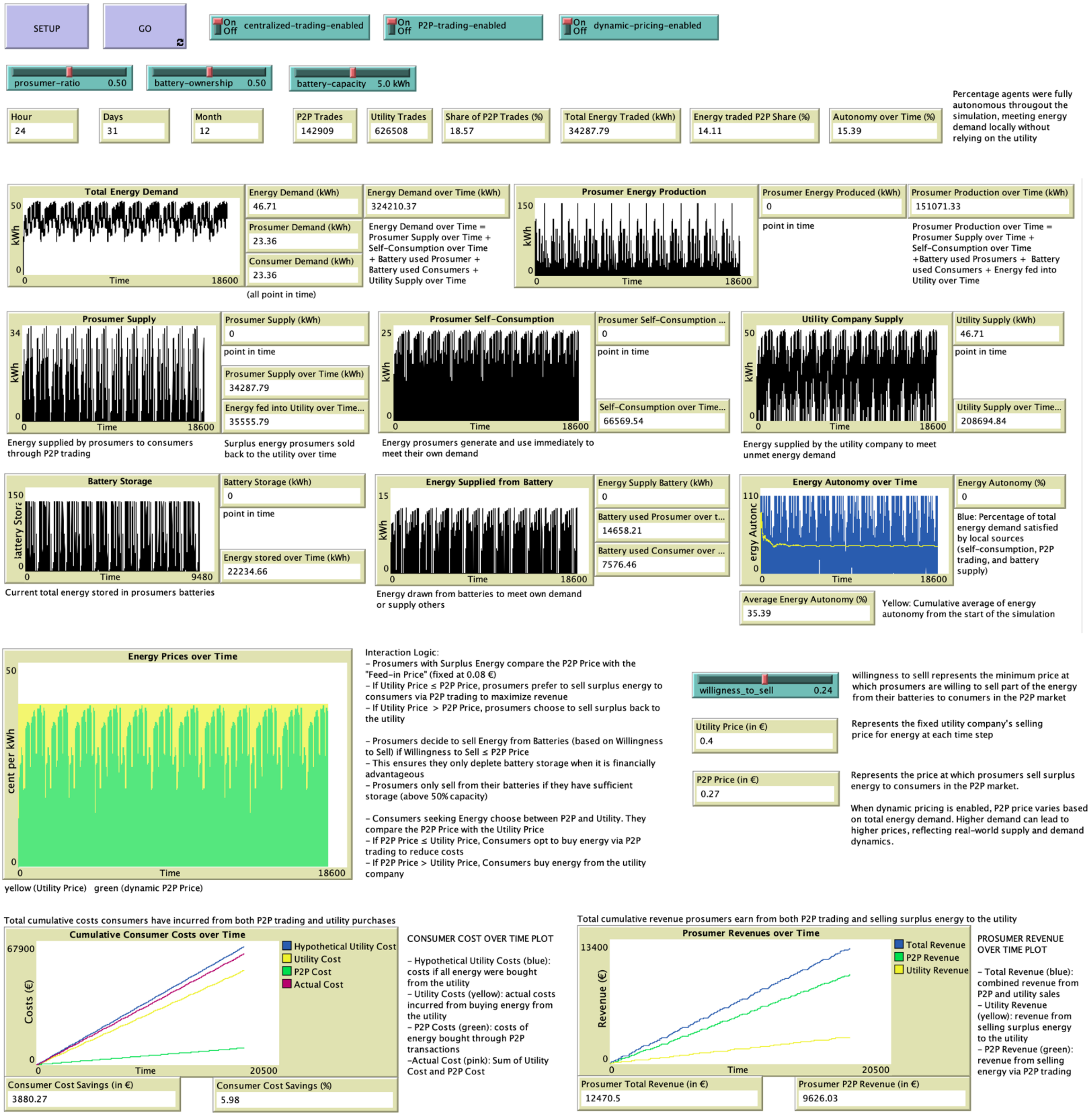
Prosumer Total Revenue (in €) 9409.75

Prosumer P2P Revenue (in €) 6452.13

PROSUMER REVENUE OVER TIME PLOT

- Total Revenue (blue): combined revenue from P2P and utility sales
- Utility Revenue (yellow): revenue from selling surplus energy to the utility
- P2P Revenue (green): revenue from selling energy via P2P trading

Figure 4: Simulation Interface - Alternative Scenario 1



prosumer-ratio 0.50

battery-ownership 0.50

battery-capacity 5.0 kWh

Hour: 24

Days: 31

Month: 12

P2P Trades: 142909

Utility Trades: 626508

Share of P2P Trades (%): 18.57

Total Energy Traded (kWh): 34287.79

Energy traded P2P Share (%): 14.11

Autonomy over Time (%): 15.39

Percentage agents were fully autonomous througout the simulation, meeting energy demand locally without relying on the utility

Energy Demand (kWh): 46.71

Prosumer Demand (kWh): 23.36

Consumer Demand (kWh): 23.36

(all point in time)

Energy Demand over Time (kWh): 324210.37

Energy Demand over Time = Prosumer Supply over Time + Self-Consumption over Time + Battery used Prosumer + Battery used Consumers + Utility Supply over Time

Prosumer Energy Produced (kWh): 0

point in time

Prosumer Production over Time (kWh): 151071.33

Prosumer Production over Time = Prosumer Supply over Time + Self-Consumption over Time +Battery used Prosumers + Battery used Consumers + Energy fed into Utility over Time

Prosumer Supply (kWh): 0

point in time

Prosumer Supply over Time (kWh): 34287.79

Energy fed into Utility over Time...: 35555.79

Energy supplied by prosumers to consumers through P2P trading

Surplus energy prosumers sold back to the utility over time

Prosumer Self-Consumption ...: 0

point in time

Self-Consumption over Time...: 66569.54

Energy prosumers generate and use immediately to meet their own demand

Utility Supply (kWh): 46.71

point in time

Utility Supply over Time...: 208694.84

Energy supplied by the utility company to meet unmet energy demand

Battery Storage (kWh): 0

point in time

Energy stored over Time (kWh): 22234.66

Current total energy stored in prosumers batteries

Energy Supply Battery (kWh): 0

Battery used Prosumer over t...: 14658.21

Battery used Consumer over ...: 7576.46

Energy drawn from batteries to meet own demand or supply others

Energy Autonomy (%): 0

Blue: Percentage of total energy demand satisfied by local sources (self-consumption, P2P trading, and battery supply)

Average Energy Autonomy (%): 35.39

Yellow: Cumulative average of energy autonomy from the start of the simulation

yellow (Utility Price) green (dynamic P2P Price)

Interaction Logic:
- Prosumers with Surplus Energy compare the P2P Price with the "Feed-in Price" (fixed at 0.08 €)
- If Utility Price ≤ P2P Price, prosumers prefer to sell surplus energy to consumers via P2P trading to maximize revenue
- If Utility Price > P2P Price, prosumers choose to sell surplus back to the utility

- Prosumers decide to sell Energy from Batteries (based on Willingness to Sell) if Willingness to Sell ≤ P2P Price
- This ensures they only deplete battery storage when it is financially advantageous
- Prosumers only sell from their batteries if they have sufficient storage (above 50% capacity)

- Consumers seeking Energy choose between P2P and Utility. They compare the P2P Price with the Utility Price
- If P2P Price ≤ Utility Price, Consumers opt to buy energy via P2P trading to reduce costs
- If P2P Price > Utility Price, Consumers buy energy from the utility company

willigness_to_sell 0.24

willingness to selll represents the minimum price at which prosumers are willing to sell part of the energy from their batteries to conumers in the P2P market

Utility Price (in €): 0.4

Represents the fixed utility company's selling price for energy at each time step

P2P Price (in €): 0.27

Represents the price at which prosumers sell surplus energy to consumers in the P2P market.

When dynamic pricing is enabled, P2P price varies based on total energy demand. Higher demand can lead to higher prices, reflecting real-world supply and demand dynamics.

Total cumulative costs consumers have incurred from both P2P trading and utility purchases

CONSUMER COST OVER TIME PLOT

- Hypothetical Utility Costs (blue): costs if all energy were bought from the utility
- Utility Costs (yellow): actual costs incurred from buying energy from the utility
- P2P Costs (green): costs of energy bought through P2P transactions
-Actual Cost (pink): Sum of Utility Cost and P2P Cost

Consumer Cost Savings (in €): 3880.27

Consumer Cost Savings (%): 5.98

Total cumulative revenue prosumers earn from both P2P trading and selling surplus energy to the utility

PROSUMER REVENUE OVER TIME PLOT

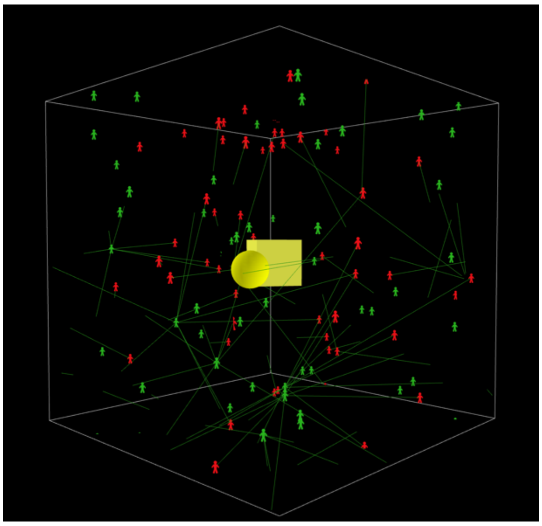
- Total Revenue (blue): combined revenue from P2P and utility sales
- Utility Revenue (yellow): revenue from selling surplus energy to the utility
- P2P Revenue (green): revenue from selling energy via P2P trading

Prosumer Total Revenue (in €): 12470.5

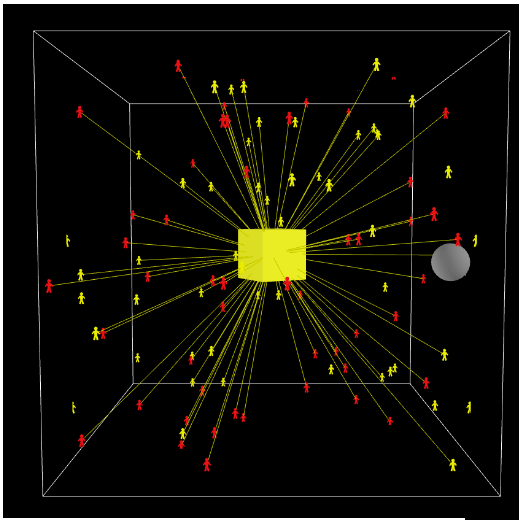
Prosumer P2P Revenue (in €): 9626.03

Figure 5: Simulation Interface - Alternative Scenario 2

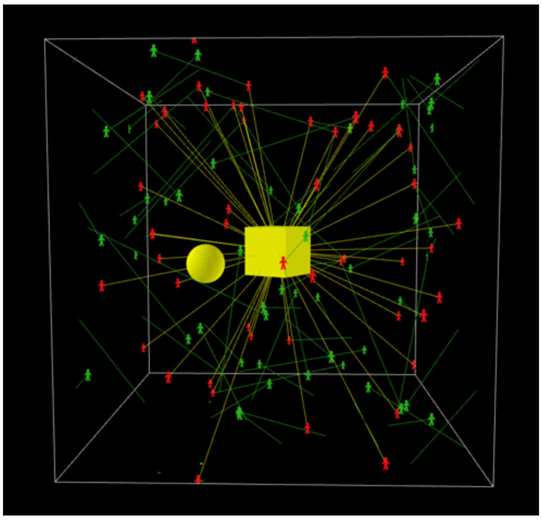
Visual Representation of Trading Scenarios in the 3D Simulation Environment



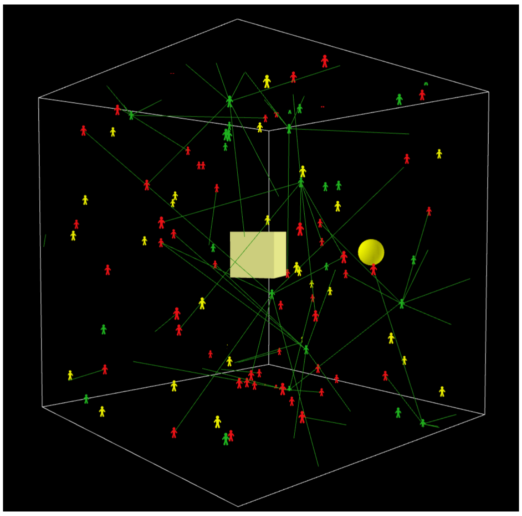
Agents Trading P2P



Centralized Trading



P2P and Centralized Trading



Energy State Agents

Table 1: The ODD+D Protocol

Structural Elements	Guiding Questions and Answers
I) OVERVIEW	
I.i Purpose	
I.i.a What is the purpose of the study?	Compare the performance of centralized and blockchain-based P2P energy trading systems, focusing on financial benefits and autonomy
I.i.b For whom is the model designed?	The model is designed for end-users, energy utility companies, and policymakers interested in the impacts of localized P2P trading
I.ii Entities, state variables, and scales	
I.ii.a What kinds of entities are in the model?	The model includes consumers, prosumers, and a utility company
I.ii.b By what attributes (i.e., state variables and parameters) are these entities characterised?	Consumers: energy demand and willingness to trade; Prosumers: production rates, consumption levels, and surplus energy storage; Utility: energy supply role
I.ii.c What are the exogenous factors / drivers of the model?	Exogenous factors include energy production variability, time of day, and weather conditions
I.ii.d If applicable, how is space included in the model?	Spatial aspects are minimal as it focuses on interactions within a microgrid community where agents interact locally
I.ii.e What are the temporal and spatial resolutions and extents of the model?	Temporal resolution is hourly with 24-hour cycles, days, and months
I.iii Process overview and scheduling	
I.iii.a What entity does what, and in what order?	Energy production starts the process, followed by storage by prosumers, trading surplus with consumers, and utility supply if necessary
II) DESIGN CONCEPTS	
II.i Theoretical and Empirical Background	
II.i.a Which general concepts, theories or hypotheses are underlying the model's design?	The model is based on principles of energy market microeconomics and blockchain technology for decentralization and transparency. Agents operate on a bounded rationality principle
II.i.b On what assumptions is/are the agents' decision model(s) based?	Agents prioritize using stored energy before buying; prosumers only trade surplus once their batteries are full, considering fees and pricing
II.i.c Why is/are certain decision model(s) chosen?	Decision models are chosen to simulate the most realistic trading behavior possible
II.i.d If the model/submodel is based on empirical data, where do the data come from?	Empirical data for production comes from the EU PVGIS system, while demand data is from household averages of Baden-Württemberg (Germany)
II.i.e At which level of aggregation were the data available?	The data were available at the household level for energy consumption and community-level aggregation (radiation data for hourly resolution for Stuttgart) for production
II.ii Individual Decision-Making	
II.ii.a What are the subjects and objects of the decision-making?	Subjects are prosumers and consumers; the object is energy allocation (whether to trade, store, or consume)
II.ii.b What is the basic rationality behind agent decision-making?	Agents aim to minimize costs, increase autonomy, and increase revenue / cost savings
II.ii.c How do agents make their decisions?	Agents follow predefined trading rules, diffrent for each scenario dispalyed (explained in more detail in the Simulation section)
II.ii.d Do the agents adapt their behaviour to changing endogenous and exogenous state variables?	Yes, they adjust to production and demand fluctuations
II.ii.e Do social norms or cultural values play a role in the decision-making process?	No, cultural norms are not considered, focusing on economic and tech factors
II.ii.f Do spatial aspects play a role in the decision process?	Spatial aspects are implicit via local community setup
II.ii.g Do temporal aspects play a role in the decision process?	Yes, time impacts production and demand with peak and off-peak hours
II.ii.h To what extent and how is uncertainty included in the agents' decision rules?	Uncertainty is included via stochastic variations in production (e.g. weather changes)
II.iii Learning	
II.iii.a Is individual learning included in the decision process?	No, agents have fixed rules without adapting based on outcomes
II.iii.b Is collective learning implemented in the model?	No, there is no collective learning mechanism
II.iv Individual Sensing	
II.iv.a What endogenous and exogenous state variables are individuals assumed to sense and consider in their decisions?	Prosumers perceive battery levels, energy production status, self-consumption needs and potential demand from consumers. Consumers sense current energy demand and prices
II.iv.b What state variables of which other individuals can an individual perceive?	They perceive available community energy and set prices
II.iv.c What is the spatial scale of sensing?	Within the microgrid community
II.iv.d Are the mechanisms by which agents obtain information modelled explicitly?	Yes, agents access energy and price information locally
II.iv.e Are the costs for cognition and gathering information explicitly included in the model?	No, information-gathering costs are excluded

II.v Individual Prediction	
II.v.a Which data do the agents use to predict future conditions?	Agents use recent production and consumption patterns for decisions
conditions or consequences of their decisions?	Simple rules; no advanced predictive models
II.v.c Might agents be erroneous in the prediction process, and how is it implemented?	Yes, predictions can be inaccurate due to variable production
II.vi Interaction	
II.vi.a Are interactions among agents and entities assumed as direct or indirect?	Direct interactions during trading
II.vi.b On what do the interactions depend?	Dependent on energy availability
II.vi.c If the interactions involve communication, how are such communications represented?	-
II.vi.d If a coordination network exists, how does it affect the agent behaviour?	Blockchain coordinates trades, ensuring transparency and security
II.vii Collectives	
II.vii.a Do the individuals form or belong to aggregations?	Yes, they form a community network / microgrid collective
II.vii.b How are collectives represented?	Represented as a shared trading network
II.viii Heterogeneity	
II.viii.a Are the agents heterogeneous?	Yes, production capacities and energy demands vary among agents
II.viii.b Are the agents heterogeneous in their decision-making?	Yes
II.ix Stochasticity	
II.ix.a What processes (including initialisation) are modelled by assuming they are random or partly random?	Energy production varies stochastically due to weather
II.x Observation	
II.x.a What data are collected from the ABM for testing, understanding and analysing it, and how and when are they collected?	Data on energy traded, financial benefits, and autonomy levels
II.x.b What key results, outputs or characteristics of the model are emerging from the individuals?	Emergent efficient energy use, autonomy, reduced grid dependence
III) DETAILS	
III.i Implementation Details	
III.i.a How has the model been implemented?	Implemented in NetLogo
III.i.b Is the model accessible, and if so where?	Accessible through academic or research platforms
III.ii Initialisation	
III.ii.a What is the initial state of the model world?	Initial state includes starting energy levels and production capabilities
III.ii.b Is the initialisation always the same, or is it allowed to vary among simulations?	Varies depending on random production initialization
III.ii.c Are the initial values chosen arbitrarily or based on data?	Values are based on data from energy datasets
III.iii Input Data	
III.iii.a Does the model use input from external sources such as data files or other models?	Yes, data was drawn from external sources
III.iv Submodels	
III.iv.a What are the submodels that represent the processes listed in 'Process overview and scheduling'?	Submodels include energy production, consumption, trading, and storage logic
III.iv.b What are the model parameters, their dimensions and reference values?	Parameters include production rates, storage capacities, trading prices
III.iv.c How were the submodels designed or chosen, and how were they parameterised and then tested?	-

Table 4: The ODD+D protocol including the guiding questions (Original questions (Grimm et al. 20006) and newly proposed questions (in bold print) to present a comprehensive model description)

Application of ODD+D Protocol to our ABM Simulation

I. Overview

I.i.Purpose

The purpose of this Agent-Based Model (ABM) is to compare a blockchain-based Peer-to-Peer (P2P) energy trading system with a conventional centralized energy trading system. The model aims to analyze financial benefits for participants, such as cost savings and revenue generation, while evaluating energy autonomy within the community. By simulating interactions within a community, the findings are intended to provide actionable insights for end-users, utility companies, and policymakers. These insights highlight the potential advantages of decentralized energy trading systems over conventional centralized approaches.

I.ii.Entities, State Variables, and Scales

The model consists of two types of human agents: prosumers and consumers, alongside a utility company. Prosumers are characterized by their ability to produce and consume energy, their battery storage levels, energy surplus, self-consumption, energy demand, and trading decisions. Their behaviors are influenced by the availability of dynamic pricing. Consumers, who only consume energy, are characterized by their energy demand and their willingness to trade or purchase energy based on prevailing dynamic prices. The utility company acts as an external supplier with fixed pricing and surplus buyback rates. The model operates on an hourly time scale and simulates energy trading activities over a period of one year, equivalent to 8760 time steps. Spatial considerations are minimal, as the focus is on interactions within a localized microgrid community rather than geographic distribution.

I.iii.Process Overview and Scheduling

The simulation progresses through discrete time steps, each representing one hour. During each step, prosumers generate energy and decide how to allocate their surplus, either by storing it in batteries or trading it. Consumers assess their energy needs and decide whether to purchase energy from prosumers or the utility company, based on cost considerations and availability. The trading process prioritizes P2P transactions, with any remaining unmet demand being fulfilled by the utility company. At each step, battery levels, trading outcomes, energy autonomy, and financial metrics for all agents are updated.

II. Design Concepts

II.i.Theoretical and Empirical Background

The model draws on principles of energy market microeconomics, emphasizing decentralized trading and the role of blockchain technology in ensuring secure, transparent transactions. It incorporates real-world energy production and consumption data to simulate realistic variations and patterns. Agents operate on the principle of bounded rationality, making decisions based on available information and cost-benefit analysis, reflecting real-world decision-making constraints.

II.ii.Individual Decision-Making

Prosumers and consumers are the primary decision-makers in the model. Prosumers decide how to allocate their energy surplus by weighing the benefits of storage versus trading, factoring in potential profits and autonomy goals. Their decisions are influenced by dynamic pricing, which adjusts based on demand-supply conditions. Consumers decide whether to purchase energy from prosumers or the utility company by comparing prices and availability. Both types of agents aim to minimize their costs while maximizing efficiency in energy use. Prosumers' trading strategies adapt to dynamic pricing, but no long-term learning is modeled, and behaviors remain static across the simulation.

II.iii.Learning

The model does not incorporate learning processes for agents, whether individual or collective. Agent behavior is static, guided by predefined rules and heuristics.

II.iv.Individual Sensing

Prosumers perceive their battery storage levels, energy production status, self-consumption needs, and potential demand from consumers. Consumers are aware of their current energy demand and the prices of energy available for purchase. To simplify the model, information is assumed to be perfectly accurate, although real-world scenarios may include inaccuracies due to metering errors or blockchain transaction delays.

II.v.Individual Prediction

Agents use simple heuristic-based predictions to estimate their future energy needs and production, relying on recent patterns. These predictions are subject to errors arising from unexpected changes in energy consumption or production, reflecting the uncertainties of real-world scenarios.

II.vi.Interaction

Interaction occurs in the form of direct energy trading between prosumers and consumers, facilitated by blockchain technology. Blockchain enables secure, transparent, and seamless transactions through smart contracts. Indirect interaction with the utility company occurs when agents buy surplus energy or fulfill unmet demand. The model assumes trust in blockchain-based trading, eliminating the need for third-party intermediaries.

II.vii.Heterogeneity

Agents are heterogeneous in terms of their energy production capacities, battery storage levels, and consumption needs. This diversity reflects the varied roles and capabilities of real-world participants in microgrid communities.

II.viii.Stochasticity

Stochastic elements in the model include random variations in energy production, influenced by weather conditions, and fluctuations in consumer demand, reflecting daily and seasonal changes.

II.ix.Observation

The simulation collects data on key metrics, including the amount of energy traded, battery usage, reliance on utility energy, price fluctuations, and cost savings. Emergent patterns, such as increased energy autonomy, reduced reliance on the central grid, and financial savings for participants, are observed and analyzed to draw actionable conclusions.

III. Details

III.i.Implementation

The model is implemented in NetLogo, a platform well-suited for simulating complex agent interactions and providing real-time visualization of key metrics. The code includes toggles for enabling or disabling P2P trading, dynamic pricing, and centralized trading. These features allow for flexible scenario testing and customization.

III.ii.Initialization

The simulation begins with prosumers initialized with predefined energy production capacities, battery storage levels, and energy demand patterns, drawn from real-world data. Consumers are initialized with typical household energy consumption profiles. Initial conditions are consistent across simulation runs unless scenario testing requires adjustments.

III.iii.Input Data

The model uses real-world hourly solar PV energy production data for Baden-Württemberg, Germany, sourced from the EU PVGIS database. Energy consumption data aligns with observed patterns in German households, adjusted for daily and seasonal variability to ensure realistic simulation of demand.

III.iv.Submodels

The model consists of several submodels that govern its core functionalities. The energy production submodel simulates hourly solar PV output based on real-world data. The energy storage submodel manages battery charging and discharging, as well as surplus storage dynamics. The dynamic pricing submodel calculates P2P trading prices based on demand-supply ratios, influencing agent decisions. The trading mechanism submodel defines the logic for both P2P and centralized trading, prioritizing P2P transactions while ensuring unmet demand is satisfied through the utility company.

NetLogo Simulation Code with Explanation of each section

```
globals [
  willingness_to_sell
  production-data
  consumption-data
  p2p-trading-price
  total-energy-demand
  total-energy-supply
  total-energy-traded
  utility-supply
  utility_buyback_volume
  prosumer-supply
  battery-supply
  time
  sun-position
  days
  months
  total-prosumer-production
  p2p-trades-count
  utility-trades-count
  current-month
  energy-autonomy
  total-battery-storage
  month-days
  prosumer-self-consumption
  total-prosumer-demand
  total-consumer-demand
  battery-utilization
  toal-energy-traded-p2p-share
  total-p2p-supply
  total-centralized-supply
  current-price
  min-total-energy-demand
  max-total-energy-demand
  hypothetical_costs_utility_only
  total-consumer-actual-costs
  total-prosumer-revenues
  total-consumer-costs-p2p
  total-consumer-utility-spending
  total-prosumer-utility-spending
  total-prosumer-actual-costs
  total-prosumer-costs-utility
  total-prosumer-revenues-p2p
  fixed_feed_in_price
  utility-sell-price
  total-energy-fed-into-utility
  p2p-trade-share
  average_autonomy
  cumm_energy_autonomy
  battery-supplied-to-others-tick
  cumulative-total-energy-demand
  cumulative-total-prosumer-production
  cumulative-prosumer-supply
  cumulative-prosumer-self-consumption
  cumulative-utility-supply
  cumulative-battery-supply
  cumulative-local-energy-consumed
  cumulative-total-energy-supplied-by-prosumers
  cumulative-energy-autonomy
  cumulative-total-battery-storage
  cumulative-battery-charging
  battery-supply-to-others
  autonomous_ticks
]
```

- This section declares all global variables used throughout the simulation

- These variables track metrics such as energy demand, supply, trading volumes, prices, revenues, costs, and cumulative statistics over time

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

turtles-own [
  role
  energy-supply
  energy-demand
  energy-demand-initial
  energy-surplus
  battery-storage
  has-battery?
  energy-used-from-supply
  energy-from-battery
]
```

- Defines variables that each turtle (agent) possesses
- role: Indicates whether the agent is a "prosumer" or "consumer"
- energy-supply: Amount of energy produced by the agent
- energy-demand: Current energy demand of the agent
- energy-demand-initial: Initial energy demand before adjustments
- energy-surplus: Surplus energy after meeting own demand
- battery-storage: Energy stored in the agent's battery
- has-battery?: Boolean indicating if the agent owns a battery
- energy-used-from-supply: Energy used from current production to meet demand
- energy-from-battery: Energy drawn from the battery to meet demand

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

to setup
  clear-all
  set-default-shape turtles "person"

  ;; Initialize globals
  set p2p-trading-price 0.24 ;; fixed P2P trading price when dynamic is not enabled
  set total-energy-demand 0
  set total-energy-supply 0
  set total-energy-traded 0
  set utility-supply 0
  set utility_buyback_volume 0
  set prosumer-supply 0
  set battery-supply 0
  set time 0
  set sun-position 0
  set days 1
  set months 1
  set p2p-trades-count 0
  set utility-trades-count 0
  set current-month 1
  set energy-autonomy 0
  set total-battery-storage 0
  set prosumer-self-consumption 0
  set total-prosumer-demand 0
  set total-consumer-demand 0
  set battery-utilization 0
  set toal-energy-traded-p2p-share 0
  set total-p2p-supply 0
  set total-centralized-supply 0
  set hypothetical_costs_utility_only 0
  set total-prosumer-revenues 0
  set total-consumer-costs-p2p 0
```

```
  set total-consumer-actual-costs 0
  set total-prosumer-actual-costs 0
  set total-consumer-utility-spending 0
  set total-prosumer-costs-utility 0
  set total-prosumer-revenues-p2p 0
  set total-energy-fed-into-utility 0
  set battery-supply-to-others 0
  set battery-supplied-to-others-tick 0
  set autonomous_ticks 0
  set cumulative-total-energy-demand 0
  set cumulative-total-prosumer-production 0
  set cumulative-prosumer-self-consumption 0
  set cumulative-prosumer-supply 0
  set cumulative-utility-supply 0
  set cumulative-battery-supply 0
  set cumulative-local-energy-consumed 0
  set cumulative-total-energy-supplied-by-prosumers 0
  set cumulative-energy-autonomy 0
  set cumulative-total-battery-storage 0
  set cumulative-battery-charging 0
  set cumm_energy_autonomy 0
  set average_autonomy 0

  set fixed_feed_in_price 0.08  ;; Prosumers sell surplus energy to the utility at
€0.08 per kWh
  set utility-sell-price 0.40  ;; Consumers buy energy from the utility at €0.40
per kWh

  set current-price utility-sell-price

  ;; Actual number of days in each month
  set month-days [31 28 31 30 31 30 31 31 30 31 30 31]

;; Initialize energy production data for the entire year (January to December)
  set production-data [

;;January
[2.111356666 1.429585714 0]
[2.177883334 0.41282 0]
[1.63674 1.282322858 0]
[0.232083334 0.142408572 0]
[0.572616666 0.258745714 0]
[0.531606666 0.100771428 0]
[1.80718 1.793388572 0]
[0.390663334 0.506111428 0]
[1.797123334 0.810657142 0]
[0.443783334 0.466865714 0]
[0.13722 0.183717142 0]
[0.358456666 1.240862858 0]
[0.426006666 1.515485714 0]
[0.520273334 0.214957142 0]
[0.25083 0.6317 0]
[0.272553334 0.92038 0]
[2.78994 1.391694286 0]
[0.312736666 0.389388572 0]
[0.66311 0.128491428 0]
[1.33562 0.298157142 0]
[0.30376 0.15512 0]
[0.226396666 0.137128572 0]
[0.83012 0.646397142 0]
[0.341526666 0.195157142 0]
[0.206213334 0.127834286 0]
[0.311226666 0.091117142 0]
[0.473923334 0.169351428 0]
[0.444296666 0.20974 0]
[0.501893334 1.536505714 0]
```

```
[1.31874 0.249594286 0]
[1.32632 0.220791428 0]

;;February
[0.319276666 0.168582858 0]
[0.16906 0.243922858 0]
[0.42014 0.1848 0]
[2.039956666 1.342214286 0]
[0.459843334 0.207731428 0]
[0.215276666 0.178062858 0]
[3.648686666 2.803817142 0]
[3.713686666 2.87276 0]
[3.610933334 2.795228572 0]
[3.49561 2.74016 0]
[2.94844 2.159494286 0]
[0.69834 1.818357142 0]
[3.550003334 2.665017142 0]
[3.549303334 2.676908572 0]
[3.495316666 2.700768572 0]
[3.585046666 2.303325714 0]
[1.099063334 1.017594286 0]
[0.597143334 1.008482858 0]
[0.741766666 0.39576 0]
[3.457063334 2.906122858 0]
[3.888503334 2.867428572 0]
[3.136703334 1.477114286 0]
[1.171483334 0.479231428 0]
[0.689596666 0.184625714 0]
[1.695133334 1.009711428 0]
[0.655716666 0.50532 0]
[4.217296666 1.408968572 0]
[1.215266666 3.04412 0]

;;March
[4.249933334 3.215445714 0]
[2.668203334 3.127634286 0]
[2.69651 3.118117142 0]
[0.602536666 0.319717142 0]
[0.529716666 0.261097142 0]
[0.968953334 0.725197142 0]
[1.110223334 1.392222858 0]
[0.999753334 0.751408572 0]
[3.232206666 1.995122858 0]
[1.574216666 0.864537142 0]
[3.30531 2.672774286 0]
[1.369146666 1.278514286 0]
[2.047353334 2.482942858 0]
[2.900856666 1.682962858 0]
[2.372466666 0.426988572 0]
[4.47355 3.057397142 0]
[3.368523334 1.921102858 0]
[4.357496666 2.8401 0]
[2.771046666 1.736202858 0]
[1.092783334 0.886965714 0]
[2.55345 1.388185714 0]
[3.915383334 3.262954286 0]
[1.399303334 0.933294286 0]
[1.721573334 0.920608572 0]
[3.052193334 2.914031428 0]
[1.59365 1.656222858 0]
[3.8098 2.655108572 0]
[4.333073334 3.489137142 0]
[1.791623334 1.530282858 0]
[0.562303334 1.757951428 0]
[3.305986666 2.526931428 0]
```

```
;;April
[1.426233334 1.414725714 0]
[0.79958 0.608482858 0]
[1.55973 1.14826 0]
[5.391776666 3.794985714 0]
[5.19209 3.603688572 0]
[5.008356666 2.757908572 0.000152728]
[2.54181 0.609437142 0.0014]
[2.0587 0.81952 0]
[4.337886666 3.160448572 0.000767272]
[4.971996666 2.562725714 0]
[2.544826666 2.567651428 0.000416364]
[1.574403334 0.981865714 0.00176]
[1.725006666 1.157991428 0.00896]
[4.917253334 2.74296 0.00684]
[3.400993334 0.877508572 0.00888]
[0.770543334 0.424988572 0.003263636]
[0.79992 0.980774286 0.005567272]
[1.21729 0.684954286 0.011112728]
[1.88474 0.767245714 0.016263636]
[1.621573334 0.528951428 0.00772]
[5.11418 3.540065714 0.011656364]
[3.884616666 1.673957142 0.020872728]
[3.543906666 2.415022858 0.022127272]
[1.838323334 1.948408572 0.015656364]
[0.72129 1.596422858 0.024943636]
[3.348503334 2.700165714 0.02392]
[4.659676666 2.597034286 0.0016]
[1.049503334 0.658314286 0.0234]
[2.839436666 2.524002858 0.029423636]
[2.62101 2.098334286 0.008967272]

;;May
[1.726983334 1.034468572 0.030167272]
[2.588563334 2.079085714 0.034223636]
[4.795796666 3.22266 0.027687272]
[5.29539 3.408457142 0.02484]
[1.635156666 1.726131428 0.036352728]
[3.789513334 2.671197142 0.029423636]
[4.586573334 0.43884 0.001927272]
[1.060296666 1.458885714 0.039192728]
[3.88967 1.739571428 0.004936364]
[1.582573334 1.42432 0.00364]
[1.107833334 1.593485714 0.014743636]
[1.587933334 1.163902857 0.044863636]
[1.22606 1.160845714 0.011007272]
[2.268156666 1.929182857 0.014352728]
[4.395746666 1.591705714 0.013712728]
[0.853956666 0.442125714 0.041376364]
[3.756466666 1.750431428 0.039487272]
[5.26061 2.560171428 0.049863636]
[1.675756666 1.084851428 0.03452]
[3.989203334 2.82192 0.03300909]
[4.14974 1.99086 0.051334546]
[4.3159 2.371851428 0.010672728]
[1.137433334 2.44076 0.012103636]
[1.339613334 3.325682857 0.049736364]
[4.04121 2.42068 0.049632728]
[5.207573334 3.497331428 0.060383636]
[5.18329 3.552422857 0.05883091]
[5.165806666 3.519802857 0.052783636]
[5.211593334 3.55656 0.06415091]
[4.83374 3.35924 0.049334546]
[5.295536666 3.577 0.051247272]

;;June
```

```
[5.026443334 3.320028572 0.051774546]
[5.190696666 3.488768572 0.053352728]
[5.093763334 2.850634286 0.06516909]
[5.059326666 3.451065714 0.051383636]
[4.742303334 3.196991428 0.06504909]
[4.333696666 3.022634286 0.057343636]
[4.66778 3.285214286 0.045032728]
[4.60222 1.082174286 0.05372]
[5.00946 3.049414286 0.052023636]
[4.612223334 2.536911428 0.050376364]
[5.017026666 3.366302857 0.052]
[3.86382 3.542165714 0.047943636]
[5.305183334 3.65284 0.05187091]
[4.885523334 3.146277142 0.06164]
[4.52449 3.031922857 0.062583636]
[4.786966666 3.441228572 0.063663636]
[4.387063334 3.26122 0.046752728]
[4.566186666 3.068948572 0.01843091]
[3.31929 2.607288572 0.051127272]
[4.361133334 3.051082857 0.020345454]
[1.504906666 2.709834286 0.061063636]
[3.188136666 1.34104 0.05512]
[3.513693334 2.39438 0.053792728]
[4.787083334 3.46218 0.046176364]
[4.96194 3.476188572 0.049143636]
[4.13729 3.290834286 0.019254546]
[2.78077 2.185908572 0.03427091]
[3.169966666 2.682462857 0.052632728]
[4.264696666 2.20578 0.004225454]
[0.54204 1.854608572 0.051512728]

;;July
[4.301923334 1.324828572 0.043752728]
[3.575663334 3.393088572 0.035447272]
[2.317446666 2.607088572 0.01439091]
[2.25356 2.21178 0.053127272]
[3.36622 3.601551428 0.043607272]
[3.907436666 2.818511428 0.045592728]
[4.876433334 3.497925714 0.035816364]
[2.80337 3.323254286 0.036472728]
[4.58728 3.045217142 0.012367272]
[4.235036666 3.056682857 0.04748909]
[4.425473334 2.65682 0.0092]
[4.177063334 2.520431428 0.020225454]
[4.381773334 2.9149 0.033]
[2.755586666 2.997342857 0.03375091]
[4.113723334 2.200465714 0.011]
[4.0371 2.66522 0.041927272]
[3.89981 1.699891428 0.033094546]
[3.011386666 1.279065714 0.036014546]
[4.267633334 2.65466 0.040776364]
[3.898383334 3.2846 0.032343636]
[4.323043334 0.693354286 0.038112728]
[3.358343334 2.9923 0.027552728]
[3.584186666 1.382897142 0.006607272]
[0.7563 1.543794286 0.012223636]
[2.19654 1.764525714 0.0078]
[2.025773334 2.582437142 0.02344]
[1.109036666 1.023214286 0.001872728]
[1.37129 0.63854 0.001863636]
[2.709143334 1.946674286 0.021303636]
[2.582756666 2.830194286 0.025552728]
[3.637776666 1.218525714 0.002247272]

;;August
[1.253046666 3.273937142 0.00152]
```

```
[1.16625 1.220097142 0.027376364]
[3.066096666 3.091342857 0.027576364]
[2.242896666 1.317457142 0.025456364]
[2.699433334 2.391125714 0.026992728]
[1.54048 1.106145714 0.028176364]
[2.723716666 1.234614286 0.026023636]
[3.8864 2.357057142 0.0154]
[1.608393334 1.9162 0.020696364]
[4.16487 3.40496 0.021672728]
[4.567803334 3.284851428 0.017943636]
[3.168456666 1.235748572 0.001496364]
[2.27172 2.269368572 0.020176364]
[3.79113 1.430025714 0.003023636]
[2.879343334 3.194965714 0.017903636]
[4.036893334 2.440228572 0.000207272]
[3.808173334 0.613085714 0.01704]
[4.544523334 3.184477142 0.015256364]
[4.373803334 2.191762857 0.01548]
[4.310986666 2.88462 0.01436]
[4.260446666 2.703057142 0.013087272]
[4.01855 2.708111428 0.012927272]
[4.21938 2.984054286 0.001863636]
[3.950526666 2.063042857 0.011552728]
[0.934073334 1.18614 0.009456364]
[1.045336666 0.330077142 0]
[0.807676666 0.583117142 0.000847272]
[1.08381 0.423202857 0]
[0.53451 0.579628572 0.007127272]
[3.054496666 1.339322857 0.007967272]
[2.52469 0.595334286 0.002712728]

;;September
[1.785156666 2.05748 0.00608]
[4.326033334 2.413502857 0.003063636]
[3.355436666 2.264594286 0.004776364]
[4.710096666 3.141748572 0.003872728]
[4.750666666 3.064577142 0.003583636]
[4.675543334 3.104888572 0.003152728]
[4.64357 3.051337142 0.002392728]
[4.482746666 2.934211428 0.001287272]
[4.34003 2.8751 0.000352728]
[4.356893334 2.851602857 0.000367272]
[4.332416666 2.781954286 0]
[3.599006666 2.22474 0]
[1.08118 1.086611428 0]
[1.862606666 2.908905714 0]
[4.461833334 2.873465714 0]
[4.264886666 2.457074286 0]
[3.851143334 2.337794286 0]
[0.480333334 1.16834 0]
[2.938536666 2.428848572 0]
[3.960516666 2.783331428 0]
[2.609303334 1.68286 0]
[1.18587 2.005308572 0]
[3.643183334 0.863768572 0]
[4.326413334 2.702254286 0]
[4.54008 2.720948572 0]
[4.414743334 2.643908572 0]
[4.399736666 2.667228572 0]
[3.75248 2.266851428 0]
[3.2187 1.677508572 0]
[2.56991 1.607511428 0]

;;October
[4.221346666 2.482317142 0]
[4.14046 2.361502857 0]
```

```
[2.598436666 0.967708572 0]
[3.629443334 1.875142857 0]
[2.042463334 0.551177142 0]
[4.082323334 2.095242857 0]
[1.60836 0.969448572 0]
[1.914473334 1.814708572 0]
[0.5413 1.074691428 0]
[3.15483 1.574497142 0]
[3.99168 2.312476 0]
[0.277693334 0.286251428 0]
[3.762546666 2.159811428 0]
[1.31418 0.261148572 0]
[1.99888 1.34776 0]
[3.91408 2.37488 0]
[1.640216666 0.575717142 0]
[2.384006666 0.251942857 0]
[1.33967 0.219834286 0]
[0.40948 0.327748572 0]
[2.748153334 0.813937142 0]
[3.728033334 1.41304 0]
[0.574286666 0.18098 0]
[0.23816 0.156402857 0]
[1.109286666 0.232934286 0]
[0.25643 0.793171428 0]
[1.461296666 1.665111428 0]
[2.330416666 0.329245714 0]
[0.61239 1.315717142 0]
[2.473576666 0.41228 0]
[1.311056666 0.291068572 0]

;;November
[3.362863334 0.951505714 0]
[0.219663334 0.074951428 0]
[0.868516666 0.609251428 0]
[2.605826666 0.194165714 0]
[2.17524 0.389248572 0]
[2.595533334 0.651514286 0]
[1.191553334 0.601934286 0]
[3.055523334 0.373837142 0]
[0.83971 0.243382857 0]
[1.9535 1.555585714 0]
[0.606686666 0.77352 0]
[1.463773334 0.311531428 0]
[0.214456666 0.1719 0]
[0.39253 0.345802857 0]
[1.339126666 1.398117142 0]
[0.430936666 0.19834 0]
[0.424496666 0.208197142 0]
[1.05365 0.24556 0]
[1.709033334 1.466897142 0]
[2.456593334 1.780642857 0]
[0.642283334 0.496734286 0]
[0.170543334 0.06638 0]
[0.775106666 0.241371428 0]
[0.785386666 0.162951428 0]
[1.33041 0.61488 0]
[1.018703334 0.77152 0]
[0.642676666 0.094711428 0]
[0.561146666 0.062894286 0]
[0.28301 1.728331428 0]
[1.55669 0.634465714 0]

;;December
[0.26227 0.061048572 0]
[0.400663334 0.06722 0]
[1.317916666 0.804848572 0]
```

```
[0.255993334 0.077462857 0]
[1.985633334 1.146942857 0]
[0.186223334 0.061334286 0]
[3.085893334 1.403348572 0]
[0.16692 0.109508572 0]
[0.775733334 0.408434286 0]
[1.054973334 0.554425714 0]
[0.783553334 1.445917142 0]
[0.240883334 0.094974286 0]
[0.9955 0.61454 0]
[0.757723334 0.155685714 0]
[0.6974 0.130125714 0]
[0.525106666 0.58456 0]
[2.724906666 1.146111428 0]
[2.949906666 1.514205714 0]
[2.12712 0.480368572 0]
[1.893643334 0.112754286 0]
[0.125576666 0.093771428 0]
[0.18342 0.077777142 0]
[0.184993334 0.102594286 0]
[0.093893334 0.305434286 0]
[1.149893334 0.412457142 0]
[0.399606666 0.174125714 0]
[2.53323 0.7544 0]
[0.143703334 0.192997142 0]
[1.519186666 0.823528572 0]
[1.579556666 1.011551428 0]
[0.615456666 0.063562857 0]
  ]

  ;; Initialize energy consumption data for the entire year (January to December)
  set consumption-data [
;;January
[0.217443333 0.245447273 0.232732066]
[0.321963939 0.349523636 0.278377851]
[0.341392121 0.35466 0.298019504]
[0.36807697 0.388610909 0.300244463]
[0.360065152 0.370585455 0.277493884]
[0.26430303 0.295536364 0.251224793]
[0.271330606 0.298245455 0.256467273]
[0.26005 0.296614545 0.271503967]
[0.412603333 0.429970909 0.335346281]
[0.431914848 0.434425455 0.336860826]
[0.432294545 0.443067273 0.340758843]
[0.438227576 0.441632727 0.338065289]
[0.43650303 0.42612 0.324152066]
[0.320402727 0.345370909 0.271112893]
[0.280330909 0.32512 0.30705124]
[0.441992727 0.453518182 0.35869157]
[0.448099697 0.456378182 0.362274876]
[0.456024545 0.466952727 0.370059339]
[0.461743333 0.469343636 0.370799835]
[0.455833636 0.444067273 0.352230744]
[0.354936061 0.375283636 0.310474876]
[0.31087 0.342127273 0.323608264]
[0.459802424 0.473203636 0.371652562]
[0.462050909 0.470912727 0.368198843]
[0.460462121 0.473741818 0.368778512]
[0.463105152 0.479576364 0.377201653]
[0.464496667 0.459563636 0.349343967]
[0.345774545 0.36984 0.306205455]
[0.304253939 0.331123636 0.333336529]
[0.457413939 0.47434 0.37259438]
[0.462676667 0.467138182 0.369284132]

;;February
```

```
[0.463790303 0.471392727 0.36816876]
[0.460262727 0.467067273 0.365364132]
[0.454641515 0.448858182 0.335748926]
[0.333291212 0.344098182 0.297387769]
[0.300102727 0.337216364 0.317577851]
[0.449045758 0.459598182 0.373135868]
[0.456389394 0.442870909 0.380846281]
[0.457615455 0.441205455 0.381790413]
[0.459467273 0.438965455 0.379895207]
[0.454980909 0.423009091 0.351467107]
[0.343076364 0.333909091 0.291751901]
[0.288179394 0.297003636 0.316116529]
[0.437896667 0.430258182 0.362057355]
[0.447429394 0.430605455 0.362075868]
[0.447622424 0.431085455 0.36675719]
[0.436220909 0.422861818 0.344043636]
[0.427517576 0.422732727 0.316483306]
[0.322309697 0.329469091 0.274109587]
[0.279045455 0.298701818 0.296530413]
[0.402606061 0.386898182 0.337471736]
[0.408895455 0.388521818 0.332326446]
[0.417798182 0.411096364 0.33102595]
[0.414397879 0.413989091 0.334876529]
[0.4165 0.41232 0.322476694]
[0.323247273 0.338963636 0.294330909]
[0.298257273 0.325981818 0.322076364]
[0.442228182 0.44432 0.36886876]
[0.456979091 0.439481818 0.368046116]

;;March
[0.448233333 0.431327273 0.361424463]
[0.442711818 0.428976364 0.35481438]
[0.433193939 0.410834545 0.336973058]
[0.336583333 0.341009091 0.293445785]
[0.293637273 0.319514545 0.313013388]
[0.438611515 0.444512727 0.357080992]
[0.445433333 0.434981818 0.355694876]
[0.441398788 0.440650909 0.34337719]
[0.421204848 0.414245455 0.33354595]
[0.425519394 0.425434545 0.333547107]
[0.327326364 0.320985455 0.287717355]
[0.283939091 0.301796364 0.292542149]
[0.420714848 0.404010909 0.329760165]
[0.425230909 0.430429091 0.351327107]
[0.428728788 0.426292727 0.34986]
[0.414433939 0.399305455 0.33662595]
[0.411222424 0.388581818 0.301660661]
[0.298168182 0.291614545 0.253609421]
[0.258624545 0.280425455 0.278219339]
[0.409077879 0.405149091 0.323903306]
[0.403622121 0.399118182 0.321761653]
[0.398575758 0.384552727 0.314866942]
[0.405639394 0.414 0.316645289]
[0.420330909 0.407270909 0.306587273]
[0.307853636 0.309478182 0.268618347]
[0.274722424 0.291309091 0.289959669]
[0.40853697 0.399623636 0.334480826]
[0.420536667 0.386347273 0.328862314]
[0.424711212 0.397432727 0.317369587]
[0.434521818 0.408849091 0.321319669]
[0.414402121 0.375538182 0.294608595]

;;April
[0.331488182 0.323609091 0.264152231]
[0.281412727 0.290634545 0.294293884]
[0.429594242 0.410263636 0.341692562]
```

```
[0.422017273 0.383483636 0.337733223]
[0.419446364 0.380347273 0.332957025]
[0.404752727 0.371974545 0.275790744]
[0.280101818 0.277416364 0.240892562]
[0.270335758 0.28128 0.23441438]
[0.241444848 0.227483636 0.224028926]
[0.248633636 0.244609091 0.259406116]
[0.399553636 0.377934545 0.307713058]
[0.423444848 0.394794545 0.314422645]
[0.409256061 0.393216364 0.319638512]
[0.39051303 0.354156364 0.286530248]
[0.296265455 0.295172727 0.248156364]
[0.271920303 0.286881818 0.277624628]
[0.415271818 0.400530909 0.323358347]
[0.429017273 0.397316364 0.32026562]
[0.427082727 0.413961818 0.327193884]
[0.414344848 0.408770909 0.320718017]
[0.405855758 0.365798182 0.272890083]
[0.29547 0.281061818 0.226612562]
[0.254420303 0.254552727 0.269734876]
[0.409830909 0.388312727 0.312660496]
[0.429068182 0.399036364 0.321982645]
[0.407760606 0.381452727 0.310419339]
[0.403859697 0.38158 0.303159008]
[0.413547273 0.376434545 0.279813719]
[0.300726364 0.280521818 0.23740876]
[0.254763939 0.246738182 0.217078678]
[0.254763939 0.246738182 0.217078678]

;;May
[0.251677576 0.253176364 0.253073719]
[0.397175758 0.370534545 0.300249091]
[0.396352727 0.364021818 0.295060992]
[0.380804242 0.362027273 0.292680992]
[0.403798182 0.370503636 0.26544]
[0.286991515 0.266109091 0.223364793]
[0.249940303 0.261521818 0.252557686]
[0.407253636 0.384190909 0.295144298]
[0.402160606 0.381945455 0.298416364]
[0.405497273 0.383265455 0.296980496]
[0.401244242 0.3838 0.297882975]
[0.398889697 0.368041818 0.269059174]
[0.300077273 0.287607273 0.224380661]
[0.255432121 0.253230909 0.257236694]
[0.387437273 0.37226 0.294201322]
[0.409828788 0.395729091 0.301094876]
[0.388141515 0.365821818 0.266295041]
[0.263089697 0.260398182 0.238327438]
[0.31892 0.319729091 0.242237025]
[0.284115152 0.270383636 0.217387603]
[0.237891818 0.24898 0.25727719]
[0.380131818 0.37132 0.291773884]
[0.407552727 0.375343636 0.290963967]
[0.39977 0.373112727 0.295701983]
[0.386476364 0.3657 0.293356694]
[0.380010909 0.350725455 0.260776033]
[0.281972727 0.267316364 0.210792562]
[0.22918 0.22664 0.198852066]
[0.233916667 0.237210909 0.237402975]
[0.367597576 0.351925455 0.28348843]
[0.373901818 0.35794 0.287416529]

;;June
[0.371604545 0.364687273 0.289388099]
[0.364441212 0.339554545 0.255989421]
[0.274966364 0.262616364 0.210917521]
```

```
[0.232518788 0.235730909 0.239590909]
[0.365035152 0.3529 0.278860331]
[0.371977879 0.356441818 0.277094711]
[0.365217576 0.349396364 0.248807769]
[0.250877879 0.262349091 0.224708099]
[0.303774545 0.299890909 0.23518843]
[0.266939697 0.268696364 0.217557686]
[0.241718485 0.247629091 0.247169421]
[0.382815152 0.375510909 0.291704463]
[0.388847879 0.373909091 0.297759174]
[0.391548182 0.376250909 0.294500992]
[0.38897303 0.370947273 0.293512893]
[0.383300909 0.353601818 0.261059504]
[0.283217879 0.273245455 0.222031901]
[0.24158697 0.255012727 0.263734545]
[0.397924545 0.388532727 0.301084463]
[0.405974545 0.399638182 0.306201983]
[0.421236667 0.394074545 0.303029421]
[0.405970303 0.406596364 0.302434711]
[0.39408303 0.362298182 0.260938017]
[0.286336061 0.27102 0.22324562]
[0.24438697 0.252232727 0.258016529]
[0.400745758 0.388158182 0.313809421]
[0.421149697 0.377492727 0.298329587]
[0.402489394 0.37882 0.293266446]
[0.391070909 0.384143636 0.301374876]
[0.410363333 0.368436364 0.263279835]

;;July
[0.28936303 0.281501818 0.22635686]
[0.252950303 0.253963636 0.253841983]
[0.394061818 0.377203636 0.295043636]
[0.404156667 0.385105455 0.304946612]
[0.404735758 0.371994545 0.297707107]
[0.38773 0.371132727 0.294421157]
[0.391340303 0.364112727 0.270407107]
[0.298677273 0.287692727 0.229656694]
[0.259443333 0.270265455 0.266932562]
[0.416572121 0.40226 0.309567769]
[0.417159697 0.405730909 0.31265124]
[0.416979394 0.399416364 0.299859174]
[0.398603333 0.376607273 0.29393405]
[0.392784848 0.362656364 0.269839008]
[0.304769394 0.294845455 0.237224793]
[0.259950303 0.261729091 0.255818182]
[0.392407273 0.380890909 0.296887934]
[0.398558788 0.397027273 0.30441438]
[0.408159394 0.389394545 0.301133058]
[0.398221515 0.379569091 0.295443967]
[0.39205303 0.370585455 0.258269917]
[0.294960909 0.274654545 0.222992231]
[0.250148182 0.263289091 0.255887603]
[0.412147273 0.393781818 0.294308926]
[0.403747273 0.380887273 0.293157686]
[0.405830303 0.379994545 0.290687438]
[0.394594242 0.377363636 0.291263636]
[0.394842424 0.366729091 0.256801653]
[0.288813636 0.281103636 0.219214545]
[0.250899091 0.25306 0.246189421]
[0.364880303 0.34874 0.277225455]

;;August
[0.394097879 0.359027273 0.275902975]
[0.385568485 0.366907273 0.276473388]
[0.365824242 0.354541818 0.27386314]
[0.364330909 0.341985455 0.243431074]
```

```
[0.274323636 0.264990909 0.211799174]
[0.248402424 0.266765455 0.24469686]
[0.357587576 0.341241818 0.26266314]
[0.35302697 0.3433 0.267916033]
[0.368365455 0.349592727 0.26875719]
[0.352445758 0.335034545 0.265583471]
[0.349713636 0.331516364 0.24309438]
[0.26845 0.271901818 0.215635868]
[0.245006364 0.248196364 0.233798843]
[0.358355455 0.349516364 0.272187769]
[0.373666364 0.345376364 0.26889719]
[0.360014242 0.350609091 0.271559504]
[0.360828788 0.353063636 0.270464959]
[0.353871212 0.330190909 0.248254711]
[0.272323333 0.276247273 0.216158843]
[0.242473636 0.253569091 0.24543157]
[0.381387576 0.370065455 0.291007934]
[0.388071515 0.371661818 0.288434711]
[0.38512303 0.367207273 0.284702149]
[0.380630303 0.371247273 0.282958512]
[0.386018182 0.351887273 0.25463686]
[0.284957273 0.27952 0.212292066]
[0.247774545 0.251901818 0.241364628]
[0.381968788 0.367127273 0.279583471]
[0.387763939 0.373129091 0.283900331]
[0.384198182 0.356330909 0.28193686]
[0.378759394 0.361930909 0.286363636]

;;September
[0.381198788 0.343458182 0.248126281]
[0.276116061 0.2681 0.215823306]
[0.23927697 0.248456364 0.246710083]
[0.378956667 0.358712727 0.28666562]
[0.38715303 0.367489091 0.292596529]
[0.385282121 0.365969091 0.290392397]
[0.385097576 0.372554545 0.292660165]
[0.385765758 0.361510909 0.264918182]
[0.295185758 0.286672727 0.227000165]
[0.253667273 0.258341818 0.256002149]
[0.393688485 0.380518182 0.298245124]
[0.406275758 0.390234545 0.303223802]
[0.412049697 0.39694 0.299189256]
[0.399466667 0.372312727 0.289688926]
[0.388226364 0.353321818 0.261517686]
[0.283519091 0.272692727 0.22199719]
[0.247331212 0.251521818 0.254517686]
[0.408426667 0.386841818 0.292406777]
[0.399252424 0.366610909 0.28855157]
[0.389477879 0.36448 0.287281157]
[0.391963939 0.376201818 0.287003471]
[0.398949091 0.356254545 0.251879669]
[0.283054545 0.273605455 0.214293719]
[0.242091818 0.23926 0.245497521]
[0.378659697 0.352427273 0.281043636]
[0.381179697 0.358350909 0.28295157]
[0.386754242 0.360925455 0.287204793]
[0.390629697 0.366225455 0.28214281]
[0.384204545 0.353830909 0.252306612]
[0.279414545 0.265274545 0.210636364]

;;October
[0.235583939 0.23784 0.221779669]
[0.306744242 0.297230909 0.228127107]
[0.257892727 0.272950909 0.252269587]
[0.383521515 0.361189091 0.283851736]
[0.397767576 0.376150909 0.286721157]
```

```
[0.392549394 0.352954545 0.260280826]
[0.290922121 0.280376364 0.217737025]
[0.247602727 0.253161818 0.253445124]
[0.405471818 0.389636364 0.292984132]
[0.403454545 0.38408 0.296130083]
[0.408235758 0.384403636 0.300984959]
[0.411835455 0.39082 0.295116529]
[0.404054848 0.369963636 0.275119669]
[0.319123636 0.306898182 0.240847438]
[0.272611818 0.279956364 0.277210413]
[0.412932121 0.379134545 0.308875868]
[0.426253333 0.399370909 0.320476198]
[0.435151818 0.418469091 0.318216529]
[0.426223636 0.412267273 0.314656364]
[0.429042727 0.3988 0.284366612]
[0.31955 0.308501818 0.246596694]
[0.273260909 0.275121818 0.27569124]
[0.424144848 0.412603636 0.313580331]
[0.434133636 0.423938182 0.322203636]
[0.43132303 0.416327273 0.315999669]
[0.436621818 0.411250909 0.311796198]
[0.421915455 0.392138182 0.286249091]
[0.314620303 0.311167273 0.246889421]
[0.269557273 0.295627273 0.268708595]
[0.396132121 0.405923636 0.308318182]
[0.405327576 0.406583636 0.288338678]

;;November
[0.273405152 0.294445455 0.267981983]
[0.401583636 0.419101818 0.310803471]
[0.402705758 0.405090909 0.303882149]
[0.311451212 0.334483636 0.258058182]
[0.268833939 0.310865455 0.282390413]
[0.412592727 0.423869091 0.32966876]
[0.421928182 0.429087273 0.336333223]
[0.426760303 0.424436364 0.336002314]
[0.430856364 0.43408 0.331444793]
[0.424352727 0.415118182 0.312843306]
[0.318230606 0.335545455 0.273915207]
[0.282776667 0.321667273 0.287249917]
[0.430746061 0.449845455 0.332570579]
[0.433026364 0.444901818 0.330795702]
[0.426442121 0.432205455 0.334370909]
[0.425078182 0.445289091 0.335679504]
[0.433172727 0.432010909 0.316424298]
[0.322778485 0.346474545 0.273698843]
[0.274631212 0.3086 0.282499174]
[0.422093636 0.432507273 0.330288926]
[0.433541818 0.442816364 0.337910248]
[0.445753636 0.460083636 0.355926281]
[0.444039697 0.447012727 0.343092562]
[0.436191212 0.438949091 0.335877355]
[0.340378182 0.364532727 0.295744793]
[0.298439697 0.334165455 0.311910744]
[0.446667879 0.464290909 0.352650744]
[0.459571212 0.477987273 0.369811736]
[0.463790303 0.462663636 0.36350595]
[0.456964242 0.466621818 0.362133719]

;;December
[0.456351212 0.459836364 0.344997025]
[0.353328182 0.387570909 0.316262314]
[0.320646667 0.348107273 0.337227603]
[0.467540606 0.48408 0.373134711]
[0.460803636 0.463218182 0.360842479]
[0.45998697 0.47348 0.364893223]
```

```
[0.452291212 0.449043636 0.357686116]
[0.451510606 0.443949091 0.321642479]
[0.32716303 0.352854545 0.278958678]
[0.284687879 0.323641818 0.290904959]
[0.42964303 0.443349091 0.335601983]
[0.438836364 0.449116364 0.333970579]
[0.436814848 0.444190909 0.336627107]
[0.435794545 0.443863636 0.337167438]
[0.430644242 0.417685455 0.313830248]
[0.321864242 0.337183636 0.291969421]
[0.293297879 0.315032727 0.302589752]
[0.42621303 0.416203636 0.339039504]
[0.426652121 0.422514545 0.325744132]
[0.412285152 0.417616364 0.317421653]
[0.400153939 0.405398182 0.299659008]
[0.36477 0.364849091 0.27359124]
[0.286022121 0.325827273 0.250932066]
[0.257947879 0.296921818 0.236640496]
[0.253476364 0.275336364 0.237312727]
[0.254123333 0.281805455 0.244349752]
[0.290485152 0.31288 0.266030083]
[0.298526667 0.327283636 0.259787934]
[0.289990909 0.313710909 0.250621983]
[0.260877273 0.287641818 0.244158843]
[0.243742121 0.293076364 0.266388182]

  ]
```

- Initializes the simulation environment including the real-life production and consumption data

```
;; Create the utility block in the middle
create-turtles 1 [
  set role "utility-company"
  set color gray
  set size 5
  set shape "square"
  setxyz 0 0 0
]

;; Create the sun icon
create-turtles 1 [
  set role "sun"
  set shape "circle"
  set color yellow
  set size 3
  setxyz 0 15 0
]
```

- Creates the utility company and sun turtles in the 3D View

```
;; Create prosumers and consumers based on prosumer-ratio slider
let num-prosumers ceiling (100 * prosumer-ratio)
let num-consumers 100 - num-prosumers

create-turtles num-prosumers [
  set role "prosumer"
  set energy-supply 0
  set energy-surplus 0
  set energy-used-from-supply 0
  set energy-from-battery 0
  set color green
  set battery-storage 0
  setxyz random-xcor random-ycor random-zcor
]
```

- Generates prosumer and consumer agents based on the prosumer-ratio

```
  ;; Deterministic battery ownership
  let total-prosumers count turtles with [role = "prosumer"]
  let num-prosumers-with-battery round (total-prosumers * battery-ownership)

  ask n-of num-prosumers-with-battery turtles with [role = "prosumer"] [
    set has-battery? true
  ]
  ask turtles with [role = "prosumer" and has-battery? != true] [
    set has-battery? false
  ]
```

- Assigns batteries to a percentage of prosumers based on the battery-ownership slider

```
  create-turtles num-consumers [
    set role "consumer"
    set energy-demand 0
    set energy-used-from-supply 0
    set energy-from-battery 0
    set color red
    setxyz random-xcor random-ycor random-zcor
  ]
```

- Positions agents randomly in the 3D space

```
  let max-individual-demand max (map [day-data -> max day-data] consumption-data)
  let min-individual-demand min (map [day-data -> min day-data] consumption-data)

  ;; Compute maximum and minimum total energy demand
  set max-total-energy-demand max-individual-demand * (count turtles with [role =
"consumer" or role = "prosumer"])
  set min-total-energy-demand min-individual-demand * (count turtles with [role =
"consumer" or role = "prosumer"])

  reset-ticks
end
```

Computes maximum and minimum total energy demand for use in dynamic pricing

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

to update_time_and_sun
  ;; Increment time and reset every 24 hours (0 to 24 inclusive, then reset to 1)
  set time time + 1
  if time > 24 [
    set time 1
    set days days + 1

    ;; If days exceed the number of days in the current month, advance to the next
month
    let days-in-month item (current-month - 1) month-days
    if days > days-in-month [
      set days 1
      set current-month current-month + 1
      if current-month > 12 [ set current-month 1 ]
      set months months + 1
    ]
  ]
  ;;
  set sun-position time
end
```

- Manages the simulation's time, advancing hours, days, and months

`Adjusts the sun-position to reflect the time of day`

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

to animate_objects
  ;; Move agents in 3D
  ask turtles [
    if role = "prosumer" or role = "consumer" [
      set xcor xcor + (random 3 - 1) * 0.1
      set ycor ycor + (random 3 - 1) * 0.1
      set zcor zcor + (random 3 - 1) * 0.1
    ]
  ]

  ;; Move the sun across the sky
  ask turtles with [role = "sun"] [
    set xcor (sin (sun-position * 15)) * 20
    set ycor (cos (sun-position * 15)) * 15
    ifelse sun-position >= 6 and sun-position <= 20 [
      set color yellow
    ] [
      set color gray
    ]
  ]
end
```

- Animates the movement of agents and the sun to enhance the visual representation
- Agents move slightly to simulate a dynamic environment
- The sun moves across the sky, changing color to represent day and night

;;

Main Simulation Loop (Go)

```
to go
  clear-links

  ;; Move time one hour ahead
  update_time_and_sun

  ;; Update energy values
  update_energy_values_and_utility_price

  ;; Initialize trading procedures
  if P2P-trading-enabled [
    p2p-trading
  ]

  ;; Centralized trading to satisfy leftover demand
  if centralized-trading-enabled [
    centralized-trading
  ]

  ;; Check if agents are fully autonomous at this tick
  if utility-supply = 0 [
    set autonomous_ticks autonomous_ticks + 1
  ]

  ;; Adjust utility block color
  adjust-utility-block-color

  ;; Update energy autonomy
  update-autonomy
```

```
  ;; Update P2P Trading Share
  let total-supply total-p2p-supply + total-centralized-supply
  ifelse total-supply > 0 [
    set toal-energy-traded-p2p-share (total-p2p-supply / total-supply) * 100
  ] [
    set toal-energy-traded-p2p-share 0
  ]

  ;; Update P2P Trade Share
  let total-trades p2p-trades-count + utility-trades-count
  ifelse total-trades > 0 [
    set p2p-trade-share (p2p-trades-count / total-trades) * 100
  ] [
    set p2p-trade-share 0
  ]

  ;; Make agents and sun move around the space
  animate_objects

  ;; Plot procedures
  plot-total-energy-demand
  plot-utility-supply
  plot-total-prosumer-production
  plot-prosumer-supply
  plot-battery-supply
  plot-prosumer-self-consumption
  plot-dynamic-price
  plot-consumer-costs
  plot-prosumer-revenues
  plot-energy-autonomy

  ;; Update cumulative totals
  set cumulative-total-energy-demand cumulative-total-energy-demand + total-energy-
demand
  set cumulative-total-prosumer-production cumulative-total-prosumer-production +
total-prosumer-production
  set cumulative-prosumer-self-consumption cumulative-prosumer-self-consumption +
prosumer-self-consumption
  set cumulative-prosumer-supply cumulative-prosumer-supply + prosumer-supply
  set cumulative-utility-supply cumulative-utility-supply + utility-supply
  set cumulative-battery-supply cumulative-battery-supply + battery-supply
  set cumulative-local-energy-consumed cumulative-local-energy-consumed +
(prosumer-self-consumption + battery-supply + prosumer-supply)
  set cumulative-total-energy-supplied-by-prosumers cumulative-prosumer-self-
consumption + cumulative-prosumer-supply + cumulative-battery-supply
  set cumulative-total-battery-storage cumulative-total-battery-storage + total-
battery-storage

  if cumulative-total-energy-demand > 0 [
    set cumulative-energy-autonomy (cumulative-local-energy-consumed / cumulative-
total-energy-demand) * 100
  ]
    set cumulative-energy-autonomy 0

  tick
end
```

- The main loop of the simulation that runs each tick
- Updates time, energy values, and executes trading procedures
- Checks for full autonomy and adjusts visual elements
- Updates plots and cumulative statistics

Advances the simulation clock

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

to update_energy_values_and_utility_price
  set utility-supply 0
  set prosumer-supply 0
  set battery-supply 0
  set prosumer-self-consumption 0
  set utility_buyback_volume 0

  ;; Retrieve current day's production and consumption data
  let current-day ((days - 1) mod 365)
  let current-day-data item current-day production-data
  let current-consumption-data item current-day consumption-data

  ;; Extract production data for different times of the day
  let morning-production item 0 current-day-data
  let afternoon-production item 1 current-day-data
  let night-production item 2 current-day-data

  ;; Extract consumption data for different times of the day
  let morning-consumption item 0 current-consumption-data
  let afternoon-consumption item 1 current-consumption-data
  let night-consumption item 2 current-consumption-data
```

- Updates energy production, consumption, and pricing based on the time of day

```
     ifelse dynamic-pricing-enabled [
        ;; Calculate current price based on total energy demand (for dynamic
    pricing)
        let normalized-demand (total-energy-demand - min-total-energy-demand) /
    (max-total-energy-demand - min-total-energy-demand)
        set normalized-demand max list 0 (min list normalized-demand 1)
        set current-price utility-sell-price
        set p2p-trading-price fixed_feed_in_price + (current-price -
    fixed_feed_in_price) * normalized-demand ;; dynamic P2P price
      ] [
        set current-price utility-sell-price
        set p2p-trading-price 0.24 ;; the setup value for the P2P trading price
      ]
```

Calculates dynamic or fixed utility and P2P trading prices

```
  ask turtles [
    ;; Consumer and prosumer demand logic based on real-time consumption data
    if role = "consumer" or role = "prosumer" [
      ifelse (time >= 7 and time <= 10) or (time >= 18 and time <= 21) [
        set energy-demand morning-consumption
      ] [
        ifelse (time >= 0 and time <= 6) or (time >= 22 and time <= 23) [
          set energy-demand night-consumption
        ] [
          set energy-demand afternoon-consumption
        ]
      ]

      ;; Store the initial energy demand before any adjustments
      set energy-demand-initial energy-demand
      ;; Calculate hypothetical cost for consumers only
      if role = "consumer" [
        let cost_utility_only energy-demand-initial * current-price
        set hypothetical_costs_utility_only hypothetical_costs_utility_only +
cost_utility_only
      ]
      ;; Set color based on energy demand
      if energy-demand > 0 [ set color red ]
```

```
   if energy-demand = 0 [ set color blue ]
  ]

  ;; Prosumer production and battery logic
  if role = "prosumer" [
   ;; Prosumers produce energy based on time of day using dataset
   ifelse time >= 6 and time < 12 [
    set energy-supply morning-production
   ] [
    ifelse time >= 12 and time < 18 [
     set energy-supply afternoon-production
    ] [
     set energy-supply night-production
    ]
   ]

   ;; First, use current production to meet own demand
   set energy-used-from-supply min list energy-supply energy-demand
   set energy-demand energy-demand - energy-used-from-supply

   ;; If demand remains, use battery storage
   ifelse energy-demand > 0 and has-battery? and battery-storage > 0 [
    set energy-from-battery min list battery-storage energy-demand
    set battery-storage battery-storage - energy-from-battery
    set energy-demand energy-demand - energy-from-battery
    set battery-supply battery-supply + energy-from-battery
   ] [
    set energy-from-battery 0
   ]

   ;; Calculate surplus energy after meeting own demand
   set energy-surplus max list 0 (energy-supply - energy-used-from-supply)

   ;; Store any surplus energy in the battery first, if there is space
   if has-battery? [
    let available-battery-space battery-capacity - battery-storage
    let energy-to-store min list available-battery-space energy-surplus
    set battery-storage battery-storage + energy-to-store
    set energy-surplus energy-surplus - energy-to-store

    ;; Update cumulative battery charging
    set cumulative-battery-charging cumulative-battery-charging + energy-to-
store
   ]

   ;; Set color based on energy surplus
   if energy-surplus > 0 [ set color green ]
   if energy-surplus = 0 [ set color yellow ]
  ]
 ]
```

- Manages prosumers' self-consumption, battery usage, and surplus energy

```
 ;; Update global variables using initial energy demand
 set total-energy-demand sum [energy-demand-initial] of turtles
 set total-prosumer-demand sum [energy-demand-initial] of turtles with [role =
"prosumer"]
 set total-consumer-demand sum [energy-demand-initial] of turtles with [role =
"consumer"]
 set total-prosumer-production sum [energy-supply] of turtles with [role =
"prosumer"]
 set total-battery-storage sum [battery-storage] of turtles with [role =
"prosumer" and has-battery?]

 ;; Calculate prosumer self-consumption
 set prosumer-self-consumption sum [energy-used-from-supply] of turtles with [role
```

```
= "prosumer"]

  ;; Update battery utilization
  set battery-utilization battery-utilization + sum [energy-from-battery] of
turtles with [role = "prosumer"]

  ;; Handle surplus energy when P2P trading is disabled (Scenario 1)
  if not P2P-trading-enabled and centralized-trading-enabled [
    ask turtles with [role = "prosumer"] [
      if energy-surplus > 0 [
        ;; Prosumers sell surplus energy to the utility
        set utility_buyback_volume utility_buyback_volume + energy-surplus

        ;; Update the total energy fed into the utility
        set total-energy-fed-into-utility total-energy-fed-into-utility + energy-
surplus

        ;; Update prosumer revenues
        let revenue energy-surplus * fixed_feed_in_price
        set total-prosumer-revenues total-prosumer-revenues + revenue
        ;; Reset surplus
        set energy-surplus 0
      ]
    ]
  ]
end
```

- Updates global metrics and handles surplus energy when P2P trading is disabled

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

to p2p-trading
  ifelse dynamic-pricing-enabled [
    dynamic-p2p-trading
  ] [
    traditional-p2p-trading
  ]
end
```

- Directs the simulation to use either dynamic or traditional P2P trading based on the dynamic-pricing-enabled setting

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

to traditional-p2p-trading
  let energy-traded 0

  ;; Prosumers trade surplus energy with consumers
  ask turtles with [role = "prosumer" and energy-surplus > 0] [
    let available-energy energy-surplus
    let demand-turtles turtles with [energy-demand > 0]

    while [available-energy > 0 and any? demand-turtles] [
      let consumer one-of demand-turtles
      let trade-amount min list available-energy [energy-demand] of consumer

      create-link-with consumer [
        set color green
      ]

      ;; Update surplus and demand
      set energy-surplus energy-surplus - trade-amount
      set available-energy available-energy - trade-amount
      ask consumer [
```

```
        set energy-demand energy-demand - trade-amount
      ]

      ;; Track prosumer supply and P2P trades
      set prosumer-supply prosumer-supply + trade-amount
      set energy-traded energy-traded + trade-amount
      set p2p-trades-count p2p-trades-count + 1

      ;; Update costs and revenues
      let revenue trade-amount * p2p-trading-price
      set total-prosumer-revenues total-prosumer-revenues + revenue
      set total-prosumer-revenues-p2p total-prosumer-revenues-p2p + revenue

      ask consumer [
        if role = "consumer" [
          let p2p_cost trade-amount * p2p-trading-price
          set total-consumer-actual-costs total-consumer-actual-costs + p2p_cost
          set total-consumer-costs-p2p total-consumer-costs-p2p + p2p_cost
        ]
      ]

      ;; Update demand-turtles
      set demand-turtles demand-turtles with [energy-demand > 0]
    ]
  ]

  if centralized-trading-enabled [
    ;; After P2P trading, prosumers sell remaining surplus to the utility
    ask turtles with [role = "prosumer" and energy-surplus > 0] [
      set utility_buyback_volume utility_buyback_volume + energy-surplus

      ;; Update the total energy fed into the utility
      set total-energy-fed-into-utility total-energy-fed-into-utility + energy-
surplus

      let revenue energy-surplus * fixed_feed_in_price
      set total-prosumer-revenues total-prosumer-revenues + revenue
      set energy-surplus 0
    ]
  ]

  ;; Update total energy traded and total P2P supply
  set total-energy-traded total-energy-traded + energy-traded
  set total-p2p-supply total-p2p-supply + energy-traded
end
```

- In traditional P2P trading, prosumers with surplus energy trade with consumers
- Trades occur at a fixed P2P trading price (0.24 €)
- Remaining surplus energy is sold back to the utility
- Updates costs, revenues, and tracking variables

```
    ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

to dynamic-p2p-trading
  let energy-traded 0
  set battery-supplied-to-others-tick 0

  ;; Prosumers decide whether to sell surplus energy via P2P or to the utility
  ask turtles with [role = "prosumer" and energy-surplus > 0] [
    ;; Prosumers compare P2P Price with Willingness to sell
    ifelse p2p-trading-price >= willingness_to_sell [
      ;; Proceed with P2P trading
      let available-energy energy-surplus
      let demand-turtles turtles with [energy-demand > 0]
```

```
      while [available-energy > 0 and any? demand-turtles] [
        let consumer one-of demand-turtles
        ask consumer [
          ;; Consumers compare P2P price with current utility price
          if p2p-trading-price <= current-price [
            ;; Proceed with P2P trade
            let trade-amount min list available-energy energy-demand

            create-link-with myself [
              set color green
            ]

            ;; Update energy variables
            set energy-demand energy-demand - trade-amount
            ask myself [
              set available-energy available-energy - trade-amount
              set energy-surplus energy-surplus - trade-amount
            ]

            ;; Update tracking variables
            set prosumer-supply prosumer-supply + trade-amount
            set energy-traded energy-traded + trade-amount
            set p2p-trades-count p2p-trades-count + 1

            ;; Update costs and revenues
            ask myself [
              let revenue trade-amount * p2p-trading-price
              set total-prosumer-revenues total-prosumer-revenues + revenue
              set total-prosumer-revenues-p2p total-prosumer-revenues-p2p + revenue
            ]

            if role = "consumer" [
              let dynamic_p2p_cost trade-amount * p2p-trading-price
              set total-consumer-actual-costs total-consumer-actual-costs +
dynamic_p2p_cost
              set total-consumer-costs-p2p total-consumer-costs-p2p +
dynamic_p2p_cost
            ]
          ]
        ]
        ;; Update demand-turtles
        set demand-turtles demand-turtles with [energy-demand > 0]
      ]

      ;; After P2P trading, sell any remaining surplus to the utility if enabled
      if centralized-trading-enabled [
        if energy-surplus > 0 [
          ;; Sell remaining surplus to the utility
          set utility_buyback_volume utility_buyback_volume + energy-surplus

          ;; Update the total energy fed into the utility
          set total-energy-fed-into-utility total-energy-fed-into-utility + energy-
surplus

          let revenue energy-surplus * fixed_feed_in_price
          set total-prosumer-revenues total-prosumer-revenues + revenue
          set energy-surplus 0
        ]
      ]
    ] [
      ;; Prosumers prefer to sell surplus back to the utility directly
      if centralized-trading-enabled [
        if energy-surplus > 0 [
          ;; Sell surplus to the utility
          set utility_buyback_volume utility_buyback_volume + energy-surplus
```

```
          ;; Update the total energy fed into the utility
          set total-energy-fed-into-utility total-energy-fed-into-utility + energy-surplus

          let revenue energy-surplus * fixed_feed_in_price
          set total-prosumer-revenues total-prosumer-revenues + revenue
          set energy-surplus 0
        ]
      ]
    ]
  ]



  ;; Prosumers with sufficient battery storage may sell from their batteries
  ask turtles with [role = "prosumer" and battery-storage > 0.5 * battery-capacity] [
    ;; Check if P2P trading is more profitable than selling to the utility
    if (p2p-trading-price >= willingness_to_sell) and (p2p-trading-price >= fixed_feed_in_price) [
      ;; Calculate maximum allowable trade from battery
      let max-allowable-trade battery-storage - 0.5 * battery-capacity
      if max-allowable-trade > 0 [
        let demand-turtles turtles with [energy-demand > 0]

        while [max-allowable-trade > 0 and any? demand-turtles] [
          let consumer one-of demand-turtles
          ask consumer [
            if p2p-trading-price <= current-price [
              ;; Proceed with P2P trade from battery
              let trade-amount min list max-allowable-trade energy-demand

              ;; Ensure trade-amount is positive
              ifelse trade-amount > 0 [

                create-link-with myself [
                  set color green
                ]
```

- Prosumers may sell energy from their batteries if conditions are favorable, maintaining at least 50% battery capacity

```
                ;; Update energy variables
                set energy-demand energy-demand - trade-amount
                ask myself [
                  set battery-storage battery-storage - trade-amount
                  set max-allowable-trade max-allowable-trade - trade-amount

                  ;; Update battery-supply-to-others
                  set battery-supplied-to-others-tick battery-supplied-to-others-tick + trade-amount
                  set battery-supply-to-others battery-supply-to-others + trade-amount
                ]

                ;; Update tracking variables
                set prosumer-supply prosumer-supply + trade-amount
                set energy-traded energy-traded + trade-amount
                set p2p-trades-count p2p-trades-count + 1

                ;; Update costs and revenues
                ask myself [
                  let revenue trade-amount * p2p-trading-price
                  set total-prosumer-revenues total-prosumer-revenues + revenue
                  set total-prosumer-revenues-p2p total-prosumer-revenues-p2p +
```

```
revenue
                ]
                if role = "consumer" [
                  let dynamic_p2p_cost trade-amount * p2p-trading-price
                  set total-consumer-actual-costs total-consumer-actual-costs +
dynamic_p2p_cost
                  set total-consumer-costs-p2p total-consumer-costs-p2p +
dynamic_p2p_cost
                ]
              ] [
                ;; If trade-amount is zero or negative, no further trading is
possible
                set max-allowable-trade 0
              ]
            ]
          ]
          ;; Update demand-turtles
          set demand-turtles demand-turtles with [energy-demand > 0]
        ]
      ]
    ]
  ]

  ;; Update total energy traded and total P2P supply
  set total-energy-traded total-energy-traded + energy-traded
  set total-p2p-supply total-p2p-supply + energy-traded
end
```

- Costs, revenues, and tracking variables are updated accordingly
- Consumers decide to buy via P2P if the price is lower than the utility price

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

to centralized-trading
  ;; Consumers who still have unmet demand buy from the utility
  ask turtles with [role = "consumer" or role = "prosumer"] [
    if energy-demand > 0 [
      let remaining-demand energy-demand
      set utility-supply utility-supply + remaining-demand

      ;; Create link with utility and visualize trade
      let utility one-of turtles with [role = "utility-company"]
      create-link-with utility [
        set color yellow
      ]

      ;; Update energy-demand
      set energy-demand 0
      set utility-trades-count utility-trades-count + 1

      ;; Calculate cost for consumers only
      if role = "consumer" [
        let utility_cost remaining-demand * current-price
        set total-consumer-actual-costs total-consumer-actual-costs +
utility_cost
        set total-consumer-utility-spending total-consumer-utility-spending +
utility_cost
      ]
      if role = "prosumer" [
        let costs remaining-demand * current-price
        set total-prosumer-actual-costs total-prosumer-actual-costs + costs
        set total-prosumer-costs-utility total-prosumer-costs-utility + costs
      ]
    ]
  ]
```

```
     ;; Update total centralized supply
     set total-centralized-supply total-centralized-supply + utility-supply
   end
```

- Handles energy purchases from the utility company for agents with unmet demand
- Visualizes trades with color links

Updates costs for consumers and prosumers

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

to adjust-utility-block-color
  ;; Calculate the ratio of P2P trades to total trades (P2P + centralized)
  let total-trades p2p-trades-count + utility-trades-count
  let p2p-ratio ifelse-value (total-trades > 0) [ p2p-trades-count / total-trades ]
[ 0 ]

  ask turtles with [role = "utility-company"] [
    ifelse p2p-ratio >= 0.99 [
      set color blue
    ] [
      ifelse p2p-ratio > 0 [
        let white-yellow-mix (1 - p2p-ratio)
        set color scale-color yellow white-yellow-mix 0 1
      ] [
        set color yellow
      ]
    ]
  ]
end
```

- Adjusts the utility company's color based on the proportion of P2P trades

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

to update-autonomy
  set average_autonomy 0

  ;; Calculate local energy consumed by prosumers and stored in batteries
  let local-energy-consumed prosumer-self-consumption + battery-supply + prosumer-
supply

  ;; If there is total energy demand, calculate autonomy as a percentage
  if total-energy-demand > 0 [
    set energy-autonomy (local-energy-consumed / total-energy-demand) * 100
  ]

  if ticks >= 1[
    set cumm_energy_autonomy cumm_energy_autonomy + energy-autonomy
    set average_autonomy cumm_energy_autonomy / ticks
  ]

end
```

- Calculates the percentage of total energy demand met by local (non-utility) sources
- Updates average autonomy over time

;;

```
to-report percentage-autonomous
  let total_simulation_time ticks
  ifelse total_simulation_time > 0 [
```

```
    report (autonomous_ticks / total_simulation_time) * 100
  ] [
    report 0
  ]
end
```

- Reports the percentage of time steps where the system was fully autonomous (no utility energy used)

  ```
  ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
  ```

```
;; Plotting procedures

to plot-total-energy-demand
  set-current-plot "Total Energy Demand"
  plot total-energy-demand
end

to plot-utility-supply
  set-current-plot "Utility Company Supply"
  plot utility-supply
end

to plot-total-prosumer-production
  set-current-plot "Prosumer Energy Production"
  plot total-prosumer-production
end

to plot-prosumer-supply
  set-current-plot "Prosumer Supply"
  plot prosumer-supply
end

to plot-battery-supply
  set-current-plot "Energy Supplied from Battery"
  plot battery-supply
end

to plot-prosumer-self-consumption
  set-current-plot "Prosumer Self-Consumption"
  plot prosumer-self-consumption
end

to plot-dynamic-price
  set-current-plot "Energy Prices over Time"

  set-current-plot-pen "Utility Price"
  plot current-price * 100

  set-current-plot-pen "P2P Trading Price"
  plot p2p-trading-price * 100
end

to plot-consumer-costs
  set-current-plot "Cumulative Consumer Costs over Time"
  set-current-plot-pen "Hypothetical Utility Cost"
  plot hypothetical_costs_utility_only
  set-current-plot-pen "Utility Cost"
  plot total-consumer-utility-spending
  set-current-plot-pen "P2P Cost"
  plot total-consumer-costs-p2p
  set-current-plot-pen "Actual Cost"
  plot total-consumer-actual-costs
end
```

```
to plot-prosumer-revenues
  set-current-plot "Prosumer Revenues over Time"
  set-current-plot-pen "Total Revenue"
  plot total-prosumer-revenues
  set-current-plot-pen "Utility Revenue"
  plot (total-prosumer-revenues - total-prosumer-revenues-p2p)
  set-current-plot-pen "P2P Revenue"
  plot total-prosumer-revenues-p2p
end

to plot-energy-autonomy
  set-current-plot "Energy Autonomy over Time"

  set-current-plot-pen "Energy Autonomy %"
  plot energy-autonomy

  set-current-plot-pen "Average Energy Autonomy %"
  plot average_autonomy
end
```

- These procedures update the plots in the simulation interface
- Provide visual representations of key metrics over time

Note:

- *The energy production and consumption data are loaded in the setup procedure and contain 3 intervals for hourly values for each day of the year*
- *The simulation includes various sliders and switches that affect agent behaviors and trading mechanisms*
- *The comments within the code help explain specific lines and logic*