PROSUMER COMMUNITY PROJECT - COMPLETE GUIDE

THE BIG PICTURE

A neighborhood where houses with solar panels can trade energy with each other instead of only buying/selling from the utility company.

PROJECT COMPONENTS

1. PROSUMERS (Producer + Consumer) - ~100 households

What they have:
- Solar panels (PV) - generates electricity during day
- Load (consumption) - electricity usage each hour
- Battery (optional) - stores excess energy

What they do each time step:
- Calculate imbalance = load - PV_generation
  - Negative (surplus): extra energy to sell
  - Positive (deficit): need to buy energy

Why: Main actors making trading decisions

2. THREE-STEP BALANCING PROCESS

STEP 1: Self-Balancing
- Use own resources first (battery storage/discharge)
- Calculate remaining imbalance
- Why: Cheapest option - use your own energy

STEP 2: Self-Organized Trading (Peer-to-Peer)
- Prosumers post buy/sell offers with prices
- Match sellers (low→high) with buyers (high→low)
- Example matching:
  Sellers: Alice 2kWh @€0.15, Bob 3kWh @€0.18
  Buyers: Charlie 2kWh @€0.22, Diana 1kWh @€0.19
  Result: Charlie↔Alice @€0.15, Diana↔Bob @€0.18
- Why: Local trading is cheaper, uses renewable better

STEP 3: Local Market (Aggregator)
- Unmatched trades go through load aggregator
- Uses D-1 market prices ± margin (e.g., ±5%)
- Why: Safety net - everyone can always trade

3. THE REGULATOR

Role: Community "manager" who sets rules

Objective (pick ONE):
- Maximize community profit
- Maximize renewable energy usage
- Maximize peer-to-peer trading

• Minimize grid dependency

Tools (nudges/rules):
• Rewards: "P2P trading gets 10% bonus"
• Penalties: "Grid-only users pay 10% extra"
• Restrictions: "Max 5 aggregator uses/day"
• Incentives: "High sharers get trade priority"

Example Strategy:
Objective: Maximize P2P trading
→ Reward: €0.02/kWh bonus for P2P trades
→ Penalty: €0.03/kWh fee for aggregator use

Why: Guides community toward collective goals

4. BLOCKCHAIN

What: Secure ledger recording all transactions

Why:
• Trust: No central authority
• Transparency: Everyone verifies trades
• Immutability: Can’t change past records

Transaction structure:
• From: Prosumer A
• To: Prosumer B
• Energy: 2 kWh
• Price: €0.15/kWh
• Timestamp: Hour 10

Block structure:
• Previous block hash: 00023abc...
• Transactions: [tx1, tx2, tx3...]
• Nonce: 12847 (for PoW)
• Hash: 000a4f2b... (must start with "000")

Consensus Mechanisms:

Proof-of-Work (PoW):
• Miners try different nonces until hash starts with "000"
• First to find it wins, adds block
• Easier to code, more computation

Proof-of-Stake (PoS):
• Validators chosen based on stake amount
• Chosen one adds block
• Faster, less computation

Requirements: ≥10 miners/validators

5. PV GENERATION FORECAST

What: Predict solar energy generation
Why: Prosumers need to know surplus/deficit for trading

Simple approach:
• Formula based on time of day
• Peak at noon, zero at night
• Add randomness for weather

Code concept:
if 6 ≤ hour ≤ 18:
    pv = max_capacity × sin((hour-6) × π/12) × random(0.8,1.0)
else:
    pv = 0

6. PRICE FORECAST

What: Predict future electricity prices
Why: Helps prosumers decide when/at what price to trade

Your advantage: You have D-1 market data!
• Use historical prices as forecast
• Prosumers reference D-1 price for P2P pricing
• Aggregator uses D-1 price ± margin

ONE TIME STEP FLOW (e.g., Hour 10)
==========

1. Each prosumer:
• Generate PV (e.g., 3 kWh solar)
• Check load (e.g., needs 2 kWh)
• Calculate imbalance: -1 kWh (surplus)

2. Self-balancing:
• Use battery if available
• Update imbalance

3. Self-organized trading:
• Surplus → post sell offers
• Deficit → post buy offers
• Match by price sorting
• Record successful transactions

4. Local market:
• Remaining imbalances → trade with aggregator
• Use D-1 price from your data

5. Blockchain:
• Collect all transactions
• Miners/validators create block
• Add block to chain

6. Regulator:
• Check rule compliance
• Apply rewards/penalties
• Update prosumer balances

→ Repeat for 24 hours (24 time steps)

DECISIONS TO MAKE BEFORE CODING
==========

1. Prosumer Setup:
   □ All same size or different PV capacities?
   □ Include batteries? (adds complexity)

2. Regulator:
   □ Which objective? (pick ONE)
   □ Which rules/incentives?

3. Blockchain:
   □ PoW or PoS?
   □ How many miners/validators? (≥10)

4. Simplification Level:
   □ Simple price matching or sophisticated?
   □ Fixed D-1 prices or add forecasting logic?

REQUIRED SIMULATION OUTPUTS
==========

After 24 hours, show:

1. Energy Flows:
   • Amount self-balanced
   • Amount P2P traded
   • Amount aggregator traded

2. Prices:
   • Average P2P price vs D-1 price
   • Price distribution over time

3. Prosumer Outcomes:
   • Who made/saved money?
   • Energy balance by prosumer

4. Regulator Success:
   • Was objective achieved?
   • Impact of rules/incentives

5. Blockchain Stats:
   • Number of blocks created
   • Transactions per block
   • Mining/validation metrics

6. Comparison:
   • Scenario WITH P2P trading
   • Scenario WITHOUT P2P trading
   • Cost/benefit analysis

SIMPLIFICATION HINTS (from instructor)
==========

• No complex optimization needed - simple sorting works
• No real networking (TCP/IP) required - all in-memory
• No GUI required - console output fine
• Equivalent mathematical operations acceptable
• Focus on concepts, not perfection

Example: Instead of optimization for cost minimization,
just sort prices cheapest→expensive and buy in order.

PRESENTATION REQUIREMENTS (< 10 slides)
==========

1. Community Layout
   • Players, roles, assumptions
   • Objectives, control strategies

2. Mathematical Models
   • Model for each player type

3. Simulation Results
   • Key metrics and outcomes

4. Discussion
   • What worked, what didn't
   • Impact of regulator strategies

5. Conclusion
   • Key findings, lessons learned

SUBMISSION (by Jan 10, 2026)
==========

Include in ZIP file:
• Presentation slides
• Source code (any language)
• Setup instructions (environment, dependencies)
• Run instructions (how to execute)
• README with project overview

Deliverable: Working demonstration required at presentation

KEY FORMULAS & CONCEPTS
==========

Prosumer Imbalance:
  imbalance = load - PV_generation

  if imbalance > 0: need to buy (deficit)
  if imbalance < 0: have to sell (surplus)

P2P Matching Condition:
  Trade occurs when: seller_price ≤ buyer_price
  Trade price: usually seller_price or midpoint

Aggregator Price:

buy_from_aggregator = D1_price × (1 + margin)
sell_to_aggregator = D1_price × (1 - margin)

Blockchain Hash Requirement (PoW):
hash(block_data + nonce) must start with "000"
Miners increment nonce until condition met

Community Profit:
Total_profit = Σ(prosumer_revenues - prosumer_costs)

Renewable Usage:
Renewable_% = (PV_energy_used) / (total_energy_consumed) × 100

P2P Trading Ratio:
P2P_ratio = (P2P_energy) / (total_energy_traded) × 100

==========
End of Guide
==========