# Energy as Currency: A Physical and Digital Framework for Thermodynamic Economies

Alejandro Luis Vaz Mayato
Independent researcher
alejandro.vaz.myt@gmail.com

December 25, 2025

## Abstract

This paper proposes a decentralized framework for modeling energy exchange in a grid of interconnected nodes. The grid is represented as a graph whose adjacency matrix is derived from a third-rank topology tensor encoding physical connection properties. Energy transfer is described in terms of exergy to account for inefficiencies in transmission. Governance of the grid is achieved through a publicly accessible server that maintains the topology, transaction records, exchange history, and smart contracts.

Nodes may dynamically update their connections, execute signed transactions, and participate in cross-transactions with non-adjacent nodes. Such cross-transactions are resolved through local elections among intermediary nodes based on reduced Katz centrality, with outcomes enforced via smart contracts. Nodes or groups of nodes may be admitted to or ejected from the grid according to their connectivity to the main cluster, defined by aggregate centrality.

The proposed framework incentivizes strong connectivity between nodes while allowing taxation-like effects to emerge naturally from cross-transactions. This model provides a foundation for studying decentralized energy exchange, governance, and incentive structures in networked systems.

# 1 Introduction

## 1.1 Motivation

In this paper we propose a framework that integrates physical energy, economics and digital accounting to manage energy grids in which energy is the currency itself. The framework handles a grid of semi-autonomous nodes that exchange energy between them and rule over common decisions.

Current economic models integrating energy fail to integrate graph theory for representing the economy, lack self-governance capabilities and don't implement blockchain technology for immutable, auditable global accounting.

This framework proposes a grid of nodes that freely exchange energy in between them, and sign transactions for promised delivery. It accounts for the physical state of the grid as well ejection of nodes, and union of new ones. It proposes a model for local elections in which Katz centrality is reduced with a matrix that models influence decay from crowding.

It allows for emergent phenomena such as taxation on cross-transactions, and strong linking between nodes for increased voting power.

## 1.2 Summary

The grid is a system of interconnected nodes that exchange energy. Since energy transfer is not perfectly efficient, exergy is used to denote usable energy, defined as energy weighted by efficiency. The physical structure of the grid is represented by a third-rank topology tensor, from which the adjacency matrix of the corresponding graph is derived.

The grid consists of six components: nodes, connections, server access clients, a server, measuring instruments, and bidirectional power control devices. The server acts as a centralization mechanism that enables self-governance of the grid.

The server exposes a public API and stores four modules: the topology tensor, the transaction log (blockchain ledger), the exchange history (blockchain ledger), and smart contracts.

Nodes may update their connection data by disconnecting freely, when existing connection data is violated, or by jointly signing a new connection object with another node. Transactions must be signed only by the sender.

Cross-transactions occur between nodes without direct connections. In such cases, intermediary nodes participate in local elections based on reduced Katz centrality, and smart contracts imposed by the involved nodes are executed to submit each node's vote.

Nodes or groups of nodes may be ejected from the grid if they are disconnected from the main cluster, defined as the connected component with the highest aggregate Katz centrality. Nodes may also join the grid through local elections analogous to those used for cross-transactions.

Overall, the grid provides a framework that rewards strong connectivity between nodes, while taxation emerges naturally from cross-transactions.

# 2 Background

## 2.1 Economy workings

In this paper, we explore a hypothetical economic framework in which energy serves as the primary currency. Energy, serving as the medium of exchange, would be stored physically and transferred over the grid, without relying on energy-backed tokens, as proposed in previous work [2]. In this framework, "money" is literally energy.

We have chosen energy as the currency rather than tokens backed by it or other representations, because work is defined as a mechanism for transferring energy. That implies that as long as there's usable energy, work can be performed. It ties real-world value (of work) to the currency itself. Traditional currencies are backed by gold or other assets that preserve value, whereas energy is inherently tied to real-world utility fundamentally [1].

There are six magnitudes to which we will refer in this paper:

- Energy ($E$): energy is the currency in our system. Strictly, electrical energy is one form of internal energy, but for our model, we consider them equivalent since we will only track the electrical component of internal energy. Energy is measured in $J$ (Joules), which is equivalent to $kg \cdot m^2 \cdot s^{-2}$ in SI units.

- Exergy ($X$): a state variable defined by the model that symbolizes the amount of usable energy on the grid. Not all energy will be circulating on the economy, since nodes can retain energy, so energy on circulation is referred to as exergy.

  $$0 \leq X \leq E$$

  It is measured on $J$. Future research ought to tighten the definition of exergy on what "energy in circulation" stands by mathematically. Although the term "exergy" originates in thermodynamics, here it is used as a model-specific quantity denoting energy on circulation.

- Usability index ($u$): is defined as the ratio between exergy and total energy. It conveys how "usable" the energy available is.

  $$u = \frac{X}{E}$$
  $$\text{where} \quad 0 \leq u \leq 1$$

  It is a unitless magnitude.

- Economic entropy ($S$): economic entropy exhibits behavior analogous to thermodynamic entropy. Since calculating real thermodynamic entropy would be nearly impossible in a decentralized grid, we will use a simpler definition:

  $$S = -E \ln u$$

  This definition is inspired by entropy-like behavior and preserves monotonicity properties analogous to the second law under certain conditions [3].

  Economic entropy has the same units as energy for our purposes. In real thermodynamics, entropy has units of $J/K$, but we can't measure a temperature factor in our decentralized grid, so we've excluded it. Future research can update this definition if needed.

- Work ($W$): work refers to a mechanism by which energy is transferred. In this framework, it is identified with net energy transfer, mathematically defined as

  $$W = \Delta E$$

  Since work derives its units from energy, its units are also $J$.

- Power ($P$): power is defined as the variation of energy per unit time. It differs from work in that the time elapsed between the two measurements is factored in. There are two ways of measuring power: over a time frame, and in an instant.
  For measuring average power over a certain time frame, we compute

  $$P_\tau = \frac{\Delta E}{\Delta t}$$

  However, this definition is narrow in the sense that it tells us the average power but not the exact power at each specific moment.
  On the other hand, instantaneous power is obtained by letting $\Delta t$ approach 0, so that $\Delta E$ also approaches 0, and the ratio becomes the slope of the tangent linear function, which represents power.

  $$P = \lim_{\Delta t \to 0} \frac{\Delta E}{\Delta t} = \frac{\mathrm{d}E}{\mathrm{d}t} = \dot{E}$$

  We will use this latter definition of power as the derivative of energy with respect to time throughout this paper. Power is measured in $J/s$, equivalent to $kg \cdot m^2 \cdot s^{-3}$, which is defined as the watt ($W$, not to be confused with work).

We'll be proposing a grid of interconnected nodes that transfer energy among themselves. On this grid, energy functions solely as the medium of exchange, though in real-world applications, its role as a commodity must also be considered.

## 2.2 Laws

The grid economy is restricted by the laws that the currency (energy) naturally imposes, the laws of thermodynamics. Rather than restrict the economy, these laws maintain it sustainable by limiting currency availability and usefulness.
The two laws we will be focusing on are:

- $1^{\underline{o}}$ law: establishes that the change in the internal energy of a closed system equals the heat added minus the work done by it.

  $$\Delta U = Q - W$$

  This law is often called the energy conservation principle. In practice, it means that the energy of a closed system stays the same unless heat or work cross the system boundary.

- $2^{\underline{o}}$ law: indicates that the entropy of any closed system that does not interact with any other external system always increases

  $$\mathrm{d}S_{\text{closed}} \geq 0$$

  It means that if our system is isolated, the entropy of it can only increase, and thus the global usability index decreases because it is now in a state with higher entropy. For that reason, entropy inversely correlates with economic utility.

These laws enforce subtle limitations within the model, arising from the laws of thermodynamics:

- Inflation and deflation (by $1^{\text{o}}$ law): energy cannot be arbitrarily created within the system, whereas fiat money can be issued independently of underlying physical value. The first law enforces that energy has to be obtained or dissipated (which means merely transferred to an outer system), and it is always backed by its ability to do work.
  Thus, inflation and deflation become controlled processes that emerge out of the grid shifting its capabilities. On our grid, the inflation/deflation rate becomes

  $$\pi_\tau = \frac{\Delta E}{E - \Delta E} \qquad \pi = \frac{\dot{E}}{E} = \frac{P}{E}$$

- Capital speculation (by $2^{\text{o}}$ law): if the economy does not suffer inflation or deflation ($\pi = 0$, $E$ is constant), speculation is only limited to the extent that economic entropy changes (economic entropy can actually decrease locally on the grid because it's an open system).
  This is because speculation only takes place when energy is less capable of useful work. The speculation restriction can thus be generalized to "speculation becomes profitable on average whenever usability decreases."

  $$\text{Average speculation profitable whenever} \qquad \Delta u < 0$$

  We can also define an adjusted inflation rate that takes exergy instead of energy:

  $$i_\tau = \frac{\Delta X}{X - \Delta X} \qquad i = \frac{\dot{X}}{X}$$

  and thus:

  $$\text{Speculation profitable whenever} \qquad \dot{u} < 0$$

  $$\dot{u} < 0$$
  $$\frac{\mathrm{d}}{\mathrm{d}t}\frac{X}{E} < 0$$
  $$\frac{\dot{X}E - X\dot{E}}{E^2} < 0$$
  $$\dot{X}E - X\dot{E} < 0$$
  $$\dot{X}E < X\dot{E}$$
  $$\frac{\dot{X}}{X} < \frac{\dot{E}}{E}$$
  $$\pi > i$$

  $$\text{Speculation profitable whenever} \qquad \pi > i$$

  In an economy in which speculation is profitable, the usability is decreasing. It doesn't necessarily have to mean that the energy or exergy on the grid is decreasing, but that energy is increasing proportionately faster than exergy.

# 3 Physical layer

## 3.1 Grid structure

The grid consists of six types of elements:

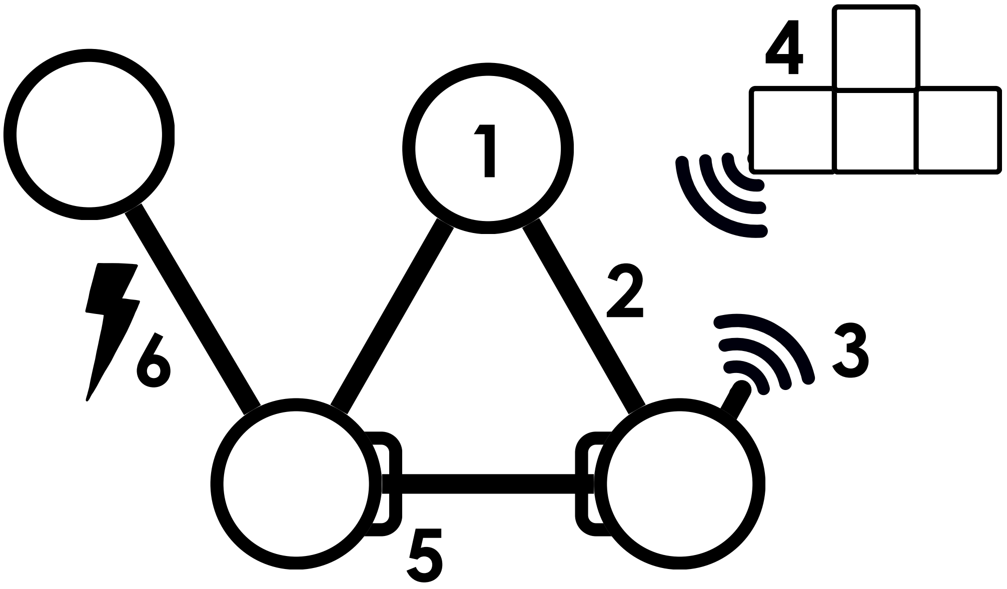

Figure 1: Diagram of grid elements

1. Nodes: nodes are the agents on the grid. They store energy, manage connections with other nodes, sign transactions, and exist independently. Each node has a unique serial ID that can't be modified.

2. Connections: connections (or "links") represent the physical wiring between two nodes. They are involved in the energy exchange between the nodes on both ends.

3. Server access client: a software client is present on each node and is in charge of reporting the node state to the server. It can interact with the server API via encrypted requests and must report frequently in order to stay connected to the grid. The node status report includes how much energy was exchanged on each link (reads from measurements).

4. Server: the server is a centralized component with the sole purpose of tracking the status of the grid, making sure each node is connected to it, and allowing nodes to create and route transactions. The server acts as a coordination and verification authority to ensure a consistent state across the grid. The server exposes a public API that nodes can reach with their clients. The server maintains four structural components:

   - Topology tensor: see 3.2.
   - Transaction log: a blockchain ledger that records all signed transactions between nodes.
   - Exchange history: a blockchain ledger that records real energy exchanged reported from nodes.
   - Smart contracts: automatic and deterministic contracts declared by nodes, which evaluate the conditions of a certain situation so that the node expresses agreement, dis-

agreement, or neutrality with decisions regarding the grid. There are two types of smart contracts: join rules and movement rules. We'll expand them on 6.2 and 5.3.

5. Bidirectional measuring instruments: nodes have a measurement instrument on each link they possess. That means that each link is delimited by two measuring instruments from the two nodes it connects. These instruments send their measurements regularly to the server so that the exchange history is kept up to date.

6. Bidirectional power control devices: nodes may send energy unilaterally at any time, provided they are not simultaneously receiving energy on the same link. Bidirectional power control devices ensure that energy exchange is bidirectional and voluntary.

## 3.2 Topology tensor

The topology tensor defines how the grid is interconnected. It is a third-rank tensor of dimensions $n \times n \times 4$ ($T \in \mathbb{R}^{n \times n \times 4}$) where $n$ is the amount on nodes on the graph:

$$T = \begin{bmatrix} T_{11} & \dots & T_{1n} \\ \vdots & \ddots & \vdots \\ T_{n1} & \dots & T_{nn} \end{bmatrix} \qquad T_{ab} = [V_n, V_e, I_M, R_w]$$

Each $T_{ab}$ represents the connection data between node $a$ and $b$. For simplicity, we will assume that $T_{abx} = T_{bax} \quad \forall x \in \{0, 1, 2, 3\}$. Theoretically, a directed graph in which links are not symmetric could be used instead of an undirected graph; future research is left to explore this alternative. Each element $T_{ab}$ is a four-dimensional vector containing the connection data, indicated by its four components:

- $V_n$: Nominal voltage. Of units $V$ (volts). Indicates the ideal voltage of the link.
- $V_e$: Voltage error. Unitless. Indicates the relative error that the actual voltage readings should accept. We can calculate minimum ($V_m$) and maximum ($V_M$) voltage as follows:

  $$V_m = V_n \cdot (1 - V_e)$$

  $$V_M = V_n \cdot (1 + V_e)$$

  In reality, for $V$ at any given time, it should hold that:

  $$V_m \leq V \leq V_M$$

- $I_M$: Maximum current. Measured in $A$ (amperes). Refers to the maximum current the instruments should read. For $I$ at any given time:

  $$I \leq I_M$$

- $R_w$: Worst-case scenario efficiency. Unitless. A precomputed value that indicates how much energy out of the sent will be delivered in the worst case. On a link, it means:

  $$X \geq E \cdot R_w$$

  $$u \geq R_w$$

  This parameter simplifies contractual expectations by providing a guaranteed minimum delivery efficiency.

### 3.3 Adjacency matrix

We can compute the adjacency matrix out of the topology tensor by specifying that for a link to truly exist $V_n > 0$ and $V_e > 0$ and $I_M > 0$, and $R_w > 0$. That, numerically, means that on the matrix $A$ of dimensions $n \times n$, element values are assigned as:

$$A_{ab} = \begin{cases} 1, & \text{if } T_{abx} > 0 \quad \forall x \in \{0, 1, 2, 3\}, \\ 0, & \text{otherwise.} \end{cases}$$

This adjacency matrix is also symmetric. We will use the adjacency matrix in situations where we want to know if nodes on the grid are connected for energy exchange or not.

# 4 Digital layer

## 4.1 Server properties

Having a centralized server is inherently risky for the framework since it becomes a single point of failure on which the system depends. For that reason, the server must be equipped with additional safety measures, both physically and digitally:

- A VPN combined with anycast routing: in order to reduce exposure, filter traffic, and improve resilience against attacks to the network.
- Not connected to the grid: the server's power supply should be administratively independent from the grid, preventing nodes from exerting economic or operational leverage over it.
- Able to clone and relocate itself: having multiple synchronized mirrors in case the main instance fails. Network traffic must also be carefully managed and redirected. Mirrors would remain synchronized via a consensus protocol, and traffic is redirected automatically to backup active instances of the server in case of failure of the main instance.
- Possess a good firewall: to defend itself against possible DoS or DDoS attacks, among other digital threats, reducing the likelihood of service disruption.

These measures ensure the server is less likely to be taken down, and in the worst case, that mirrors can keep centralized accounting operative.

## 4.2 Modules

The server holds four important modules describing the grid: the topology tensor, the transaction log, the exchange history, and the smart contracts. Out of them, the transaction log and exchange history are cryptographically signed and immutable blockchain ledgers that ensure consensus and auditability. These modules can be accessed via the server's public API. Each module is stored differently:

- Topology tensor: it is stored as shown in 3.2.
- Transaction log: a blockchain ledger that stores the transactions signed between nodes. It is thus immutable. This module exists on-server just to hold a full copy of the ledger, since smart contracts may need to access it upon execution.
- Exchange history: blockchain ledger that records the reported energy exchanges from node instruments. Nodes must report regularly in order to stay connected to the grid, ensuring inactive nodes are expelled, and exchange data isn't held by nodes.
- Smart contracts: context-aware smart contracts that nodes send to the server so that they express their stance about decisions ahead-of-time instead of holding regular elections. Each of these smart contracts is a deterministic, side-effect-free, computationally-bounded function that's evaluated when a decision needs to be taken:

**Algorithm 1** Smart Contract Decision Procedure

```
Require: Decision data decision
Require: Topology tensor topology
Require: Transaction log transactions
Require: Exchange history exchange
Ensure: Decision result in $\{\mathbf{-1}, \mathbf{0}, \mathbf{1}\}$
function SmartContract(decision, topology, transactions, exchange)
    return decision outcome
end function
```

# 5 Rules and transactions

## 5.1 Connection data update

There are three situations in which two nodes can change their connection data:

- By freely disconnecting, either of the nodes can disconnect from the link at any given time.
- If connection data has been violated (e.g., exceeded maximum allowed voltage): they must disconnect in order to hold both parties safe and the grid resilient.
- By both nodes cryptographically signing a new connection data object: and submitting it to the server via the public API.

## 5.2 Transaction signing

For a transaction to be recorded, the transaction data signed by the sender must be submitted to the server. However, the receiver signature is not mandatory, since nodes are allowed to send energy unilaterally at any time without prior notice, and it wouldn't effectively convey any consent. Transactions act as promises of delivery that become public and stored forever on the transaction log. Nevertheless, the transaction contract can be created by either of the nodes.

## 5.3 Cross-transactions

So far, we've assumed that the nodes involved in a transaction are directly connected with a link. However, some transactions might require intermediary nodes since the sender and the receiver might not be connected directly. These types of transactions are called cross-transactions.

For a cross-transaction to take place, the sender must sign the transaction contract cryptographically, and the intermediary nodes must all comply with being part of the transaction.

In order to avoid single points of failure in long-distance cross-transactions, intermediaries do not hold complete sovereignty over the decision to comply or not. Instead, deterministic, weighted, and local elections take place automatically according to movement rules for each intermediary involved.

To calculate the results of the local elections happening on each node, we must first define the $Y \in \mathbb{R}^{n\times n}$ matrix that indicates the amount of exergy transferred between node $i$ and $j$:

$$Y = \begin{bmatrix} Y_{11} & \dots & Y_{1n} \\ \vdots & \ddots & \vdots \\ Y_{n1} & \dots & Y_{nn} \end{bmatrix}$$

The data needed to populate $Y$ is obtained directly from the real exchange history stored on the server.

Then we proceed to calculate $\lambda_m$, the maximum eigenvalue of the $Y$ matrix. We will need it to assign a proper $\alpha$ value to calculate Katz centrality.

Consequently, we calculate Katz centrality, assigning values for constants heuristically, only serving as placeholders for future calibration:

$$\alpha < \frac{1}{\lambda_m} J^{-1} \to \alpha = \frac{e}{\pi\lambda_m} J^{-1} \to \alpha \approx \frac{0.865}{\lambda_m} J^{-1} \qquad \beta = 10$$

Here, $\alpha$ has units of $J^{-1}$ since it needs to cancel out joules on $Y$. Afterwards, we apply the formula to obtain $C$ (Katz centrality):

$$C = \left(I - \alpha Y^T\right)^{-1} \cdot \beta \overline{\mathbf{1}}$$

where $I$ is the identity matrix of dimensions $n \times n$, and $\overline{\mathbf{1}}$ is a vector filled with ones of length $n$.
In order to get the voting power each node has, we will now take the Katz centrality and multiply it by the reduction matrix $R \in \mathbb{R}^{n \times n}$ element-wise. The reduction matrix calculates how much the influence a node exerts over another decays with distance and the number of additional nodes nearby. Each factor in $R$ holds that:

$$R_{ij} = \frac{1}{n_d}$$

where $n_d$ refers to the number of nodes within the amount of hops from node $i$ to node $j$ around $i$. If no path exists from $i$ to $j$ we set $R_{ij} = 0$. The diagonal is always $R_{ii} = 1$. Then, the voting power each node $j$ exerts over $i$ becomes:

$$V = R \circ C^T$$

Each element of the $V$ matrix can then be rounded to integer values in order to prevent all smart contracts from being evaluated, so only those nodes whose $V_{ij} \neq 0$ have their rules computed, reducing server load significantly on large grids. Those rules are applied to get a final resolution score:

$$r_i = V_{i,:} \cdot S_{ci}$$

where $S_{ci}$ is the vector containing the result of each smart contract ($S_{ci} \in \{-1, 0, 1\}^n$).
The decision goes forward if $r_i \geq 0$, and thus a cross-transaction is signed if each intermediary goes through this process and its $r_i \geq 0$. Later on, the cross-transaction involving intermediaries is broken down into individual local transactions to be appended into the transaction log so debt stays local.
In this type of transactions, intermediaries lend infrastructure to make the transaction possible between sender and receiver, and in exchange, they benefit from delivering real efficiencies higher than those recorded on connection data, whilst keeping the difference to themselves. This behavior is analogous to taxation via fees.
The worst-case efficiency of a cross-transaction is the product of the efficiencies of all involved connections:

$$R_t = \prod R_w \quad \text{for } R_w \text{ of each link involved}$$

## 5.4 Unpaid debt

Unpaid debt doesn't get forcefully repaid, nor does the debtor get punished. Instead, the penalties are the records on the transaction log and real exchange history showing unfulfilled exchanges. Nodes with debt history will naturally be excluded for cross-transactions and have very limited trust from others, meaning that their incoming exergy will be low and thus Katz centrality and voting power on the grid.
Since the serial ID of a node cannot change even after reconnection, it ensures that debt stays permanent and publicly accessible via the server API.

# 6 Node life

## 6.1 Group ejection

At all times, the grid must be fully interconnected, so removing groups and nodes not connected to the main cluster is a task that should be run every time the adjacency matrix changes. In order to first calculate if there are disconnected components, we will calculate the multiplicity of zero-valued eigenvalues of the Laplacian by solving:

$$\det(L - \lambda I) = 0$$

$$\text{where} \qquad L = D - A$$

If the amount of zero-valued eigenvalues is different than one, the graph is not fully interconnected, and it's divided into components. To make sure that the graph becomes a single cluster once again, the group with the highest sum of Katz centrality on all its nodes is kept, and all others are ejected from the grid.

$$C = \left(I - \alpha Y^T\right)^{-1} \cdot \beta\overline{1}$$

To do so, the server must catalog each node into which component it is part of by selecting a node and traversing its connections recursively until no new nodes are connected, and then repeating the same process starting with a node that's not already in any previous groups until all nodes are cataloged. Finally, the sum of the Katz centralities of the nodes of each group is calculated, and that component which scores higher is kept, whilst all the others are ejected.

## 6.2 Node union

For a node to join, a local decision evaluated at the node where the new node wants to connect must pass. As we've explored in , the result in $r_i$ must be equal to or greater than zero:

$$r_i = \left(R \circ \left(\left(I - \alpha Y^T\right)^{-1} \cdot \beta\overline{\mathbf{1}}\right)^T\right)_{i,:} \cdot S_{ci}$$

The smart contracts to be executed (whose result is on $S_{ci}$) are provided this time with data about the decision of joining the new node that is applying for. If it is accepted, the new node is appended to the topology tensor, the adjacency matrix is recalculated, and the node is taken into consideration on the $Y$ matrix for future decisions.

# 7 Example

## 7.1 Katz centrality

For our example, we will use a 14-node grid, whose exergy exchange matrix is as follows:

$$Y = \begin{bmatrix} 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 1 & 5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 5 & 4 & 0 & 12 & 0 & 0 & 20 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 12 & 0 & 6 & 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 20 & 0 & 0 & 0 & 0 & 7 & 14 & 12 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 7 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 14 & 0 & 0 & 0 & 6 & 7 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 12 & 0 & 0 & 0 & 0 & 0 & 7 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 6 & 0 & 0 & 0 & 9 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 7 & 0 & 0 & 0 & 4 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 7 & 9 & 4 & 0 \end{bmatrix} \cdot 10^6 J$$

In order to calculate Katz centrality, first we define its constants:

$$\alpha = \frac{e}{\pi \lambda_m} \approx 2.83199 \cdot 10^{-8} J^{-1} \qquad \beta = 10$$

and then we compute it (rounded to the nearest integer for display purposes):

$$C = (I - \alpha Y)^{-1} \cdot \beta \overline{\mathbf{1}} = \begin{bmatrix} 11 \\ 26 \\ 22 \\ 101 \\ 48 \\ 18 \\ 14 \\ 120 \\ 34 \\ 68 \\ 57 \\ 30 \\ 27 \\ 32 \end{bmatrix}$$

With this information, we can represent the graph using a spring layout with node size and color as indicator of their Katz centrality and edge widths proportional to the weights on $Y$:

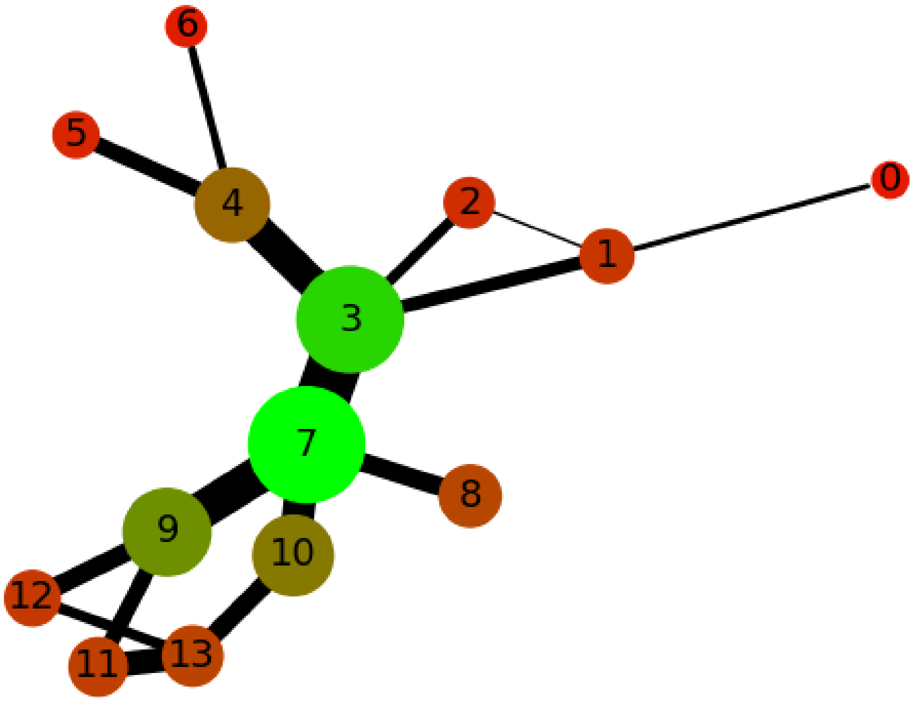

Figure 2: Grid graph.

## 7.2 Cross-transaction

We will imagine now that node 11 wants to send some energy to node 12. In order to do so, they have to find an intermediary, since they don't share a direct connection. They can either choose 9 as intermediary or choose 13. On a real scenario, they would look at the transaction log, the real exchange history, and promised efficiencies on both paths to assess which one fits more their needs, but we will continue our example with 13 as intermediary.

Node 11 creates and signs a transaction that is meant to be received by node 12, but has to first be signed by 13, where a decision takes place.

For the transaction to be successfully created, the decision on node 13 has to pass. We'll first compute the voting power matrix, using the Katz centrality from 7.1:

$$V = R \circ C^T$$

$$\approx \begin{bmatrix} 11 & \frac{26}{2} & \frac{22}{4} & \frac{101}{4} & \frac{48}{6} & \frac{18}{11} & \frac{14}{11} & \frac{120}{6} & \frac{34}{11} & \frac{68}{11} & \frac{57}{11} & \frac{30}{14} & \frac{27}{14} & \frac{32}{14} \\ \frac{11}{4} & 26 & \frac{22}{4} & \frac{101}{4} & \frac{48}{6} & \frac{18}{11} & \frac{14}{11} & \frac{120}{6} & \frac{34}{11} & \frac{68}{11} & \frac{57}{11} & \frac{30}{14} & \frac{27}{14} & \frac{32}{14} \\ \frac{11}{6} & \frac{26}{3} & 22 & \frac{101}{3} & \frac{48}{6} & \frac{18}{11} & \frac{14}{11} & \frac{120}{6} & \frac{34}{11} & \frac{68}{11} & \frac{57}{11} & \frac{30}{14} & \frac{27}{14} & \frac{32}{14} \\ \frac{11}{11} & \frac{26}{5} & \frac{22}{5} & 101 & \frac{48}{5} & \frac{18}{11} & \frac{14}{11} & \frac{120}{5} & \frac{34}{11} & \frac{68}{11} & \frac{57}{11} & \frac{30}{14} & \frac{27}{14} & \frac{32}{14} \\ \frac{11}{11} & \frac{26}{7} & \frac{22}{7} & \frac{101}{4} & 48 & \frac{18}{4} & \frac{14}{4} & \frac{120}{7} & \frac{34}{11} & \frac{68}{11} & \frac{57}{11} & \frac{30}{14} & \frac{27}{14} & \frac{32}{14} \\ \frac{11}{11} & \frac{26}{7} & \frac{22}{7} & \frac{101}{4} & \frac{48}{2} & 18 & \frac{14}{4} & \frac{120}{7} & \frac{34}{11} & \frac{68}{11} & \frac{57}{11} & \frac{30}{14} & \frac{27}{14} & \frac{32}{14} \\ \frac{11}{11} & \frac{26}{7} & \frac{22}{7} & \frac{101}{4} & \frac{48}{2} & \frac{18}{4} & 14 & \frac{120}{7} & \frac{34}{11} & \frac{68}{11} & \frac{57}{11} & \frac{30}{14} & \frac{27}{14} & \frac{32}{14} \\ \frac{11}{14} & \frac{26}{11} & \frac{22}{11} & \frac{101}{5} & \frac{48}{11} & \frac{18}{14} & \frac{14}{14} & 120 & \frac{34}{5} & \frac{68}{5} & \frac{57}{5} & \frac{30}{11} & \frac{27}{11} & \frac{32}{11} \\ \frac{11}{14} & \frac{26}{11} & \frac{22}{11} & \frac{101}{5} & \frac{48}{11} & \frac{18}{14} & \frac{14}{14} & \frac{120}{2} & 34 & \frac{68}{5} & \frac{57}{5} & \frac{30}{11} & \frac{27}{11} & \frac{32}{11} \\ \frac{11}{14} & \frac{26}{11} & \frac{22}{11} & \frac{101}{8} & \frac{48}{11} & \frac{18}{14} & \frac{14}{14} & \frac{120}{4} & \frac{34}{8} & 68 & \frac{57}{8} & \frac{30}{4} & \frac{27}{4} & \frac{32}{8} \\ \frac{11}{14} & \frac{26}{11} & \frac{22}{11} & \frac{101}{8} & \frac{48}{11} & \frac{18}{14} & \frac{14}{14} & \frac{120}{3} & \frac{34}{8} & \frac{68}{8} & 57 & \frac{30}{8} & \frac{27}{8} & \frac{32}{3} \\ \frac{11}{14} & \frac{26}{11} & \frac{22}{11} & \frac{101}{8} & \frac{48}{11} & \frac{18}{14} & \frac{14}{14} & \frac{120}{6} & \frac{34}{8} & \frac{68}{3} & \frac{57}{6} & 30 & \frac{27}{6} & \frac{32}{3} \\ \frac{11}{14} & \frac{26}{11} & \frac{22}{11} & \frac{101}{8} & \frac{48}{11} & \frac{18}{14} & \frac{14}{14} & \frac{120}{6} & \frac{34}{8} & \frac{68}{3} & \frac{57}{6} & \frac{30}{6} & 27 & \frac{32}{3} \\ \frac{11}{14} & \frac{26}{11} & \frac{22}{11} & \frac{101}{8} & \frac{48}{11} & \frac{18}{14} & \frac{14}{14} & \frac{120}{6} & \frac{34}{8} & \frac{68}{6} & \frac{57}{4} & \frac{30}{4} & \frac{27}{4} & 32 \end{bmatrix}$$

The coefficients of the denominator of the $R$ matrix are shown on each element denominator. With the voting power matrix, we can now extract the row 13 which corresponds to the voting power distribution on node 13:

$$\begin{bmatrix} \frac{11}{14} & \frac{26}{11} & \frac{22}{11} & \frac{101}{8} & \frac{48}{11} & \frac{18}{14} & \frac{14}{14} & \frac{120}{6} & \frac{34}{8} & \frac{68}{6} & \frac{57}{4} & \frac{30}{4} & \frac{27}{4} & 32 \end{bmatrix}$$

We can reason about the denominator coefficients this way:

- Nodes within 0 hops: $[13]$. Then node 13 Katz centrality gets divided by 1.
- Nodes within 1 hop: $[10, 11, 12, 13]$. Then nodes $10, 11, 12$ get divided by four since there are now four nodes within a hop from 13. 13 was already found earlier so its centrality is kept divided by 1.
- Nodes within 2 hops: $[7, 9, 10, 11, 12, 13]$. Then nodes $7, 9$ get divided by six since there are now six nodes within two hops from 13.
- Continue until all elements have been reached.

We will round now each centrality to the nearest integer in $V$ and multiply it by the $S_{ci}$ vector (randomly generated, for the sake of the example) via dot product:

$$r_i = V_{i,:} \cdot S_{ci}$$

$$r_{13} = \begin{bmatrix} 1 \\ 2 \\ 2 \\ 13 \\ 4 \\ 1 \\ 1 \\ 20 \\ 4 \\ 11 \\ 14 \\ 7 \\ 7 \\ 32 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 0 \\ 0 \\ -1 \\ 1 \\ 0 \\ 1 \\ -1 \\ 1 \\ 1 \\ -1 \\ 1 \\ 1 \\ 1 \end{bmatrix} = 1$$

The decision goes forward since $r_{13} \geq 0$ and it is subsequently broken down into two individual already-signed transactions: from 11 to 13 and from 13 to 12. As we can see, the voting power does not lean towards the nodes involved heavily, but neither to the furthermost ones. It ensures that nodes with highest Katz centrality are self-sovereign whilst those with lower get coerced only by powerful adjacent nodes.

### 7.3 Group ejection

If we suppose nodes 3 and 7 disconnect, our graph stops being fully interconnected and adopts the following structure:

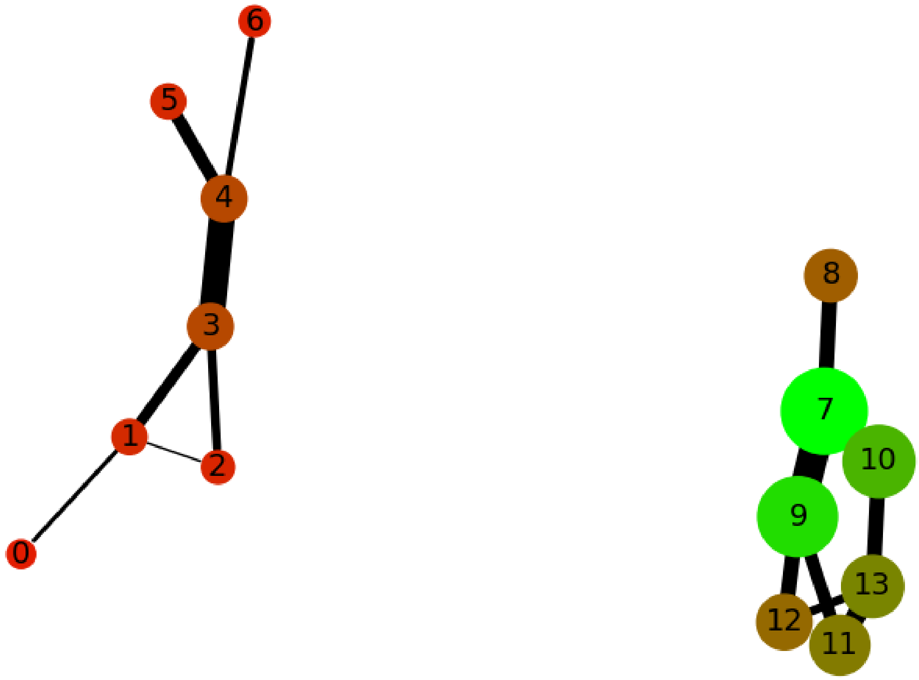

Figure 3: Disconnected graph.

We can see two big components which aren't interconnected, but we must prove mathematically that they are separated by calculating the multiplicity of zero-valued eigenvalues on the Laplacian of the adjacency matrix:

$$\det(L - \lambda I) = 0$$

$$\text{where} \qquad L = D - A$$

the solutions for $\lambda$ will repeat twice the value 0 since there are two components on the graph. In order to select one group and expel the other from the grid, we must first determine the Katz centrality of each node and then sum over the centralities of each node in each group:

$$C = \begin{bmatrix} 11 \\ 17 \\ 15 \\ 29 \\ 29 \\ 17 \\ 13 \\ 103 \\ 38 \\ 90 \\ 73 \\ 50 \\ 43 \\ 54 \end{bmatrix}$$

$$c_1 = \sum_{i=0}^{6} C_i = 134 \qquad c_2 = \sum_{i=7}^{13} C_i = 451$$

We can see that the centrality of the second group is much higher, thus it is the component kept.

# 8 Conclusion

## 8.1 Limitations

The proposed framework is subject to several physical, technical, and technological limitations. A key concern is measurement accuracy: high-precision, standardized instruments would be required at each node to ensure fairness across the system.

The stability of the grid has not been analyzed in this work and should be investigated through simulations to better understand the resulting economic dynamics. In addition, nodes would need sufficient energy storage capacity to avoid persistent push-only behavior, and decentralized elections should remain independent of large transfer hubs.

Finally, the server and its surrounding ecosystem must demonstrate adequate resilience to cyber threats, as well as sufficient computational capacity to execute smart contracts and calculate grid-level properties.

## 8.2 Emergent network behavior

Two types of behavior arise in the grid due to the nature of the framework:

- Taxation: taxation appears in cross-transactions as intermediaries deliver efficiency close to that expected whilst physical efficiency is higher, thus keeping the difference to themselves. The model rewards higher grid efficiency than expected, and thus incentivize nodes to be energetically sustainable.

  Intermediaries act as lenders of infrastructure and receive in exchange the efficiency difference, mimicking fee-like payments.

- Strong linking: nodes which actively exchange energy naturally increase their $Y$ matrix exergy and thus their voting power. In order to increase a node's power and influence on the grid, then, actively participating on the economy is key since it increases the node's entries on the $Y$ matrix.

Simulations are required to further understand the scope of these and other tendencies.

## 8.3 Directions for future research

Future research may explore and expand the framework in several directions:

- Incorporating simulations to study system behavior
- Allowing for directed connections between nodes
- Modeling real energy exchange and transaction log blockchain fields

# References

[1] Nicholas Georgescu-Roegen. *The Entropy Law and the Economic Process*. Harvard University Press, Cambridge, MA, 1971.

[2] Mihail Mihaylov, Sergio Jurado, Néstor Avellana, Kristof Van Moffaert, and Ann Nowé. Nrgcoin: Virtual currency for trading of renewable energy in smart grids. In *2014 11th International Conference on the European Energy Market (EEM)*, pages 1–5. IEEE, 2014.

[3] T. Smith and D. Foley. The thermodynamics of economic systems. *Ecological Economics*, 66(1):1–12, 2008.

```
#
#   HEAD
#

# HEAD -> MODULES
import numpy as np
from collections import deque
import networkx as nx
import matplotlib.pyplot as plt

# HEAD -> ADJACENCY MATRIX
A = np.array([
    [0, 7, 14, 12, 0, 0, 0],#7
    [7, 0, 0, 0, 0, 0, 0],#8
    [14, 0, 0, 0, 6, 7, 0],#9
    [12, 0, 0, 0, 0, 0, 7],#10
    [0, 0, 6, 0, 0, 0, 9],#11
    [0, 0, 7, 0, 0, 0, 4],#12
    [0, 0, 0, 7, 9, 4, 0]#13
]) * 10**6 # numbers on the graph are incorrect serial node ID
#A = np.array([
#    [0, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],#0
#    [2, 0, 1, 5, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],#1
#    [0, 1, 0, 4, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],#2
#    [0, 5, 4, 0, 12, 0, 0, 0, 0, 0, 0, 0, 0, 0],#3
#    [0, 0, 0, 12, 0, 6, 3, 0, 0, 0, 0, 0, 0, 0],#4
#    [0, 0, 0, 0, 6, 0, 0, 0, 0, 0, 0, 0, 0, 0],#5
#    [0, 0, 0, 0, 3, 0, 0, 0, 0, 0, 0, 0, 0, 0],#6
#    [0, 0, 0, 0, 0, 0, 0, 0, 7, 14, 12, 0, 0, 0],#7
#    [0, 0, 0, 0, 0, 0, 0, 7, 0, 0, 0, 0, 0, 0],#8
#    [0, 0, 0, 0, 0, 0, 0, 14, 0, 0, 0, 6, 7, 0],#9
#    [0, 0, 0, 0, 0, 0, 0, 12, 0, 0, 0, 0, 0, 7],#10
#    [0, 0, 0, 0, 0, 0, 0, 0, 0, 6, 0, 0, 0, 9],#11
#    [0, 0, 0, 0, 0, 0, 0, 0, 0, 7, 0, 0, 0, 4],#12
#    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 7, 9, 4, 0]#13
#]) * 10**6
#A = np.array([
#    [0, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],#0
#    [2, 0, 1, 5, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],#1
#    [0, 1, 0, 4, 0, 0, 0, 20, 0, 0, 0, 0, 0, 0],#2
#    [0, 5, 4, 0, 12, 0, 0, 0, 0, 0, 0, 0, 0, 0],#3
#    [0, 0, 0, 12, 0, 6, 3, 0, 0, 0, 0, 0, 0, 0],#4
#    [0, 0, 0, 0, 6, 0, 0, 0, 0, 0, 0, 0, 0, 0],#5
#    [0, 0, 0, 0, 3, 0, 0, 0, 0, 0, 0, 0, 0, 0],#6
#    [0, 0, 20, 0, 0, 0, 0, 0, 7, 14, 12, 0, 0, 0],#7
#    [0, 0, 0, 0, 0, 0, 0, 7, 0, 0, 0, 0, 0, 0],#8
#    [0, 0, 0, 0, 0, 0, 0, 14, 0, 0, 0, 6, 7, 0],#9
#    [0, 0, 0, 0, 0, 0, 0, 12, 0, 0, 0, 0, 0, 7],#10
#    [0, 0, 0, 0, 0, 0, 0, 0, 0, 6, 0, 0, 0, 9],#11
#    [0, 0, 0, 0, 0, 0, 0, 0, 0, 7, 0, 0, 0, 4],#12
#    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 7, 9, 4, 0]#13
#]) * 10**6

```

```
# HEAD -> CONSTANTS
n = len(A)
alpha = (np.e / np.pi / np.max(np.linalg.eigvals(A))).real
beta = 10

# HEAD -> SHOW
print(A)
print(alpha)


#
#   KATZ
#

# KATZ -> COMPUTE
C = np.linalg.inv(np.eye(n) - alpha * A.T) @ (beta * np.ones((n, 1)))

# KATZ -> SHOW
print(np.round(C, 0))


#
#   VOTING POWER
#

# VOTING POWER -> REDUCTION FN
def af(a: int, b: int) -> float:
    if a == b:
        return 1.0
    n = len(A)
    visited = [False] * n
    depth_of = {}
    depth_count = {}
    q = deque()
    q.append((a, 0))
    visited[a] = True
    depth_of[a] = 0
    depth_count[0] = 1
    while q:
        node, d = q.popleft()
        for nei in range(n):
            if A[node, nei] != 0 and not visited[nei]:
                visited[nei] = True
                depth = d + 1
                depth_of[nei] = depth
                depth_count[depth] = depth_count.get(depth, 0) + 1
                q.append((nei, depth))
    if b not in depth_of:
        return 0.0
    depth_b = depth_of[b]
    cumulative = sum(depth_count[d] for d in range(depth_b + 1))
    return 1.0 / cumulative

# VOTING POWER -> REDUCTION
```

```python
R = np.zeros((n, n))
for i in range(n):
    for j in range(n):
        R[i, j] = af(i, j)
print(R**-1)

# VOTING POWER -> CALCULATION
V = R * C.T

# VOTING POWER -> SHOW
print(np.round(V, 0))

# for decision now: get V_i, dot product with decisions S_c



#
#   GRAPH STUFF
#

G = nx.DiGraph()
for i in range(n):
    G.add_node(i)
for i in range(n):
    for j in range(n):
        if A[i, j] != 0:
            G.add_edge(i, j, weight=A[i, j]*10**-6)
pos = nx.spring_layout(G)
nx.draw_networkx_nodes(G, pos, node_color=[
    f'#{int(255*(1 - C[i,0]/np.max(C))):02x}{int(255*(C[i,0]/np.max(C))):02x}00'
    for i in range(len(C))
], node_size=[1.5*beta*C[index] for index in range(len(C))])

# Draw edges
# Draw edges with curved lines for bidirectional visibility
drawn_pairs = set()
for u, v in G.edges():
    if (v, u) in G.edges() and (v, u) not in drawn_pairs:
        # Draw u->v curved one way, v->u curved opposite
        nx.draw_networkx_edges(
            G, pos, edgelist=[(u, v)],
            width=G[u][v]['weight'],
            arrowstyle='-',
            arrowsize=10,
            connectionstyle='arc3,rad=0.0'
        )
        drawn_pairs.add((u, v))
        drawn_pairs.add((v, u))
    elif (v, u) not in G.edges():
        # Single edge or self-loop
        nx.draw_networkx_edges(
            G, pos, edgelist=[(u, v)],
            width=G[u][v]['weight'],
            arrowstyle='-',
            arrowsize=10,
            connectionstyle='arc3,rad=0.0'
        )


# Draw labels
nx.draw_networkx_labels(G, pos, font_size=14, font_color='black')

plt.axis('off')
plt.show()
```