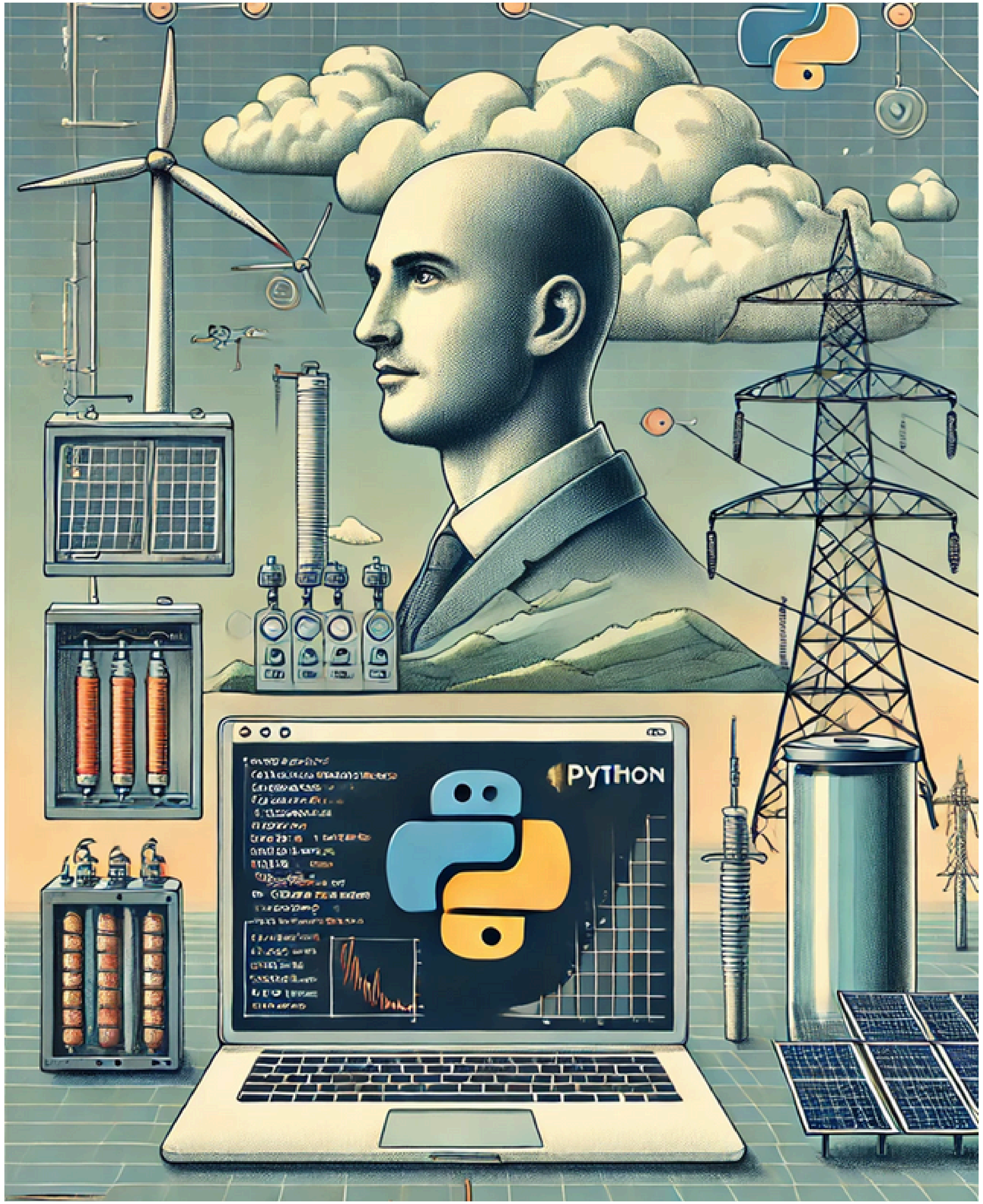


April 5, 2025

Python for Energy System Modelling

Python-Nairobi

Chrispine Tinega
ENGINEER | LEADER | CREATOR

Chrispine Tinega

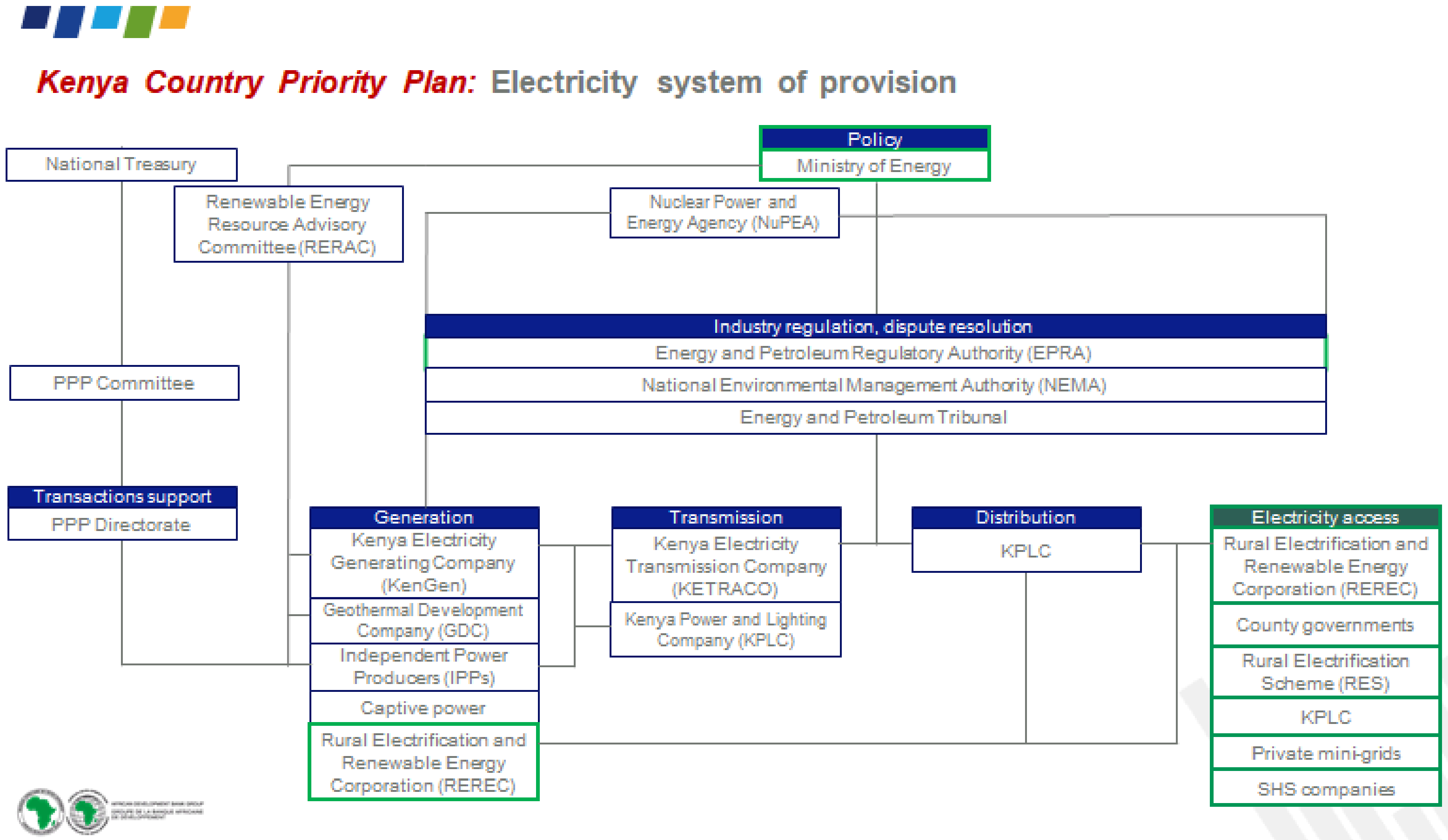
- Embedded Systems Engineer specialising in IoT for energy systems.
- Driven by a passion for leveraging technology to address real-world energy challenges.
- Mentor and speaker with hands-on experience in smart mini-grids and energy management systems.

https://chrispinetinega.com

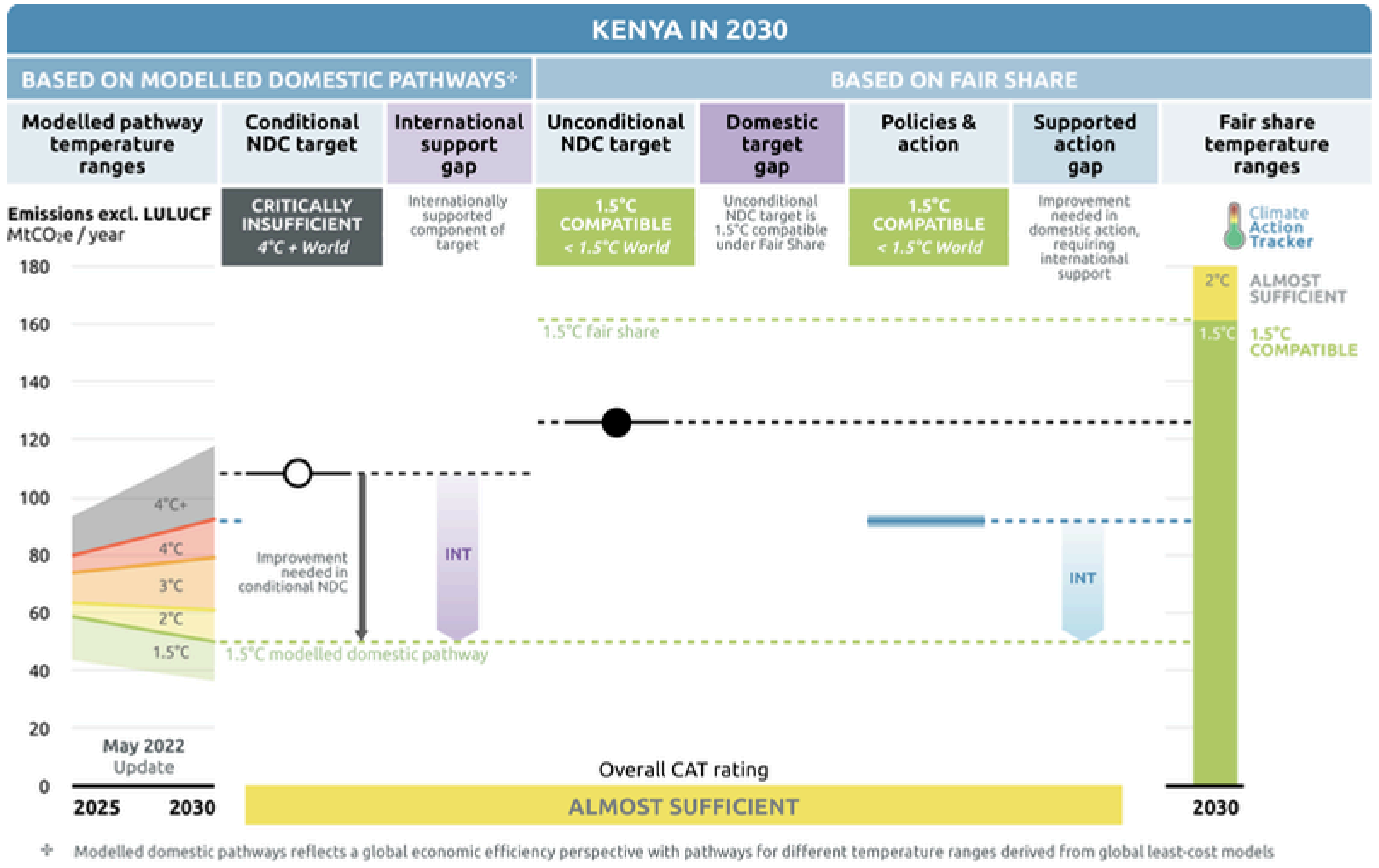
Energy Transition, Huddle?

“Kenya's energy transition faces infrastructure and funding challenges.”

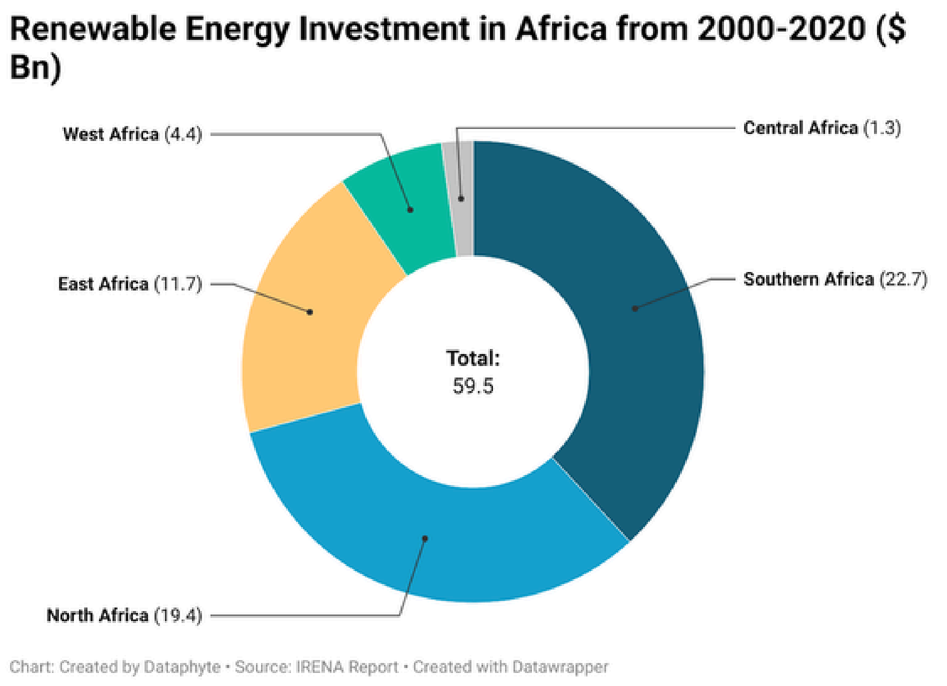
-Energy Transition and Investment Plan (ETIP)



Policies & action. (n.d.). Climateactiontracker.Org. Retrieved 4 April 2025, from https://climateactiontracker.org/countries/kenya/policies-action/



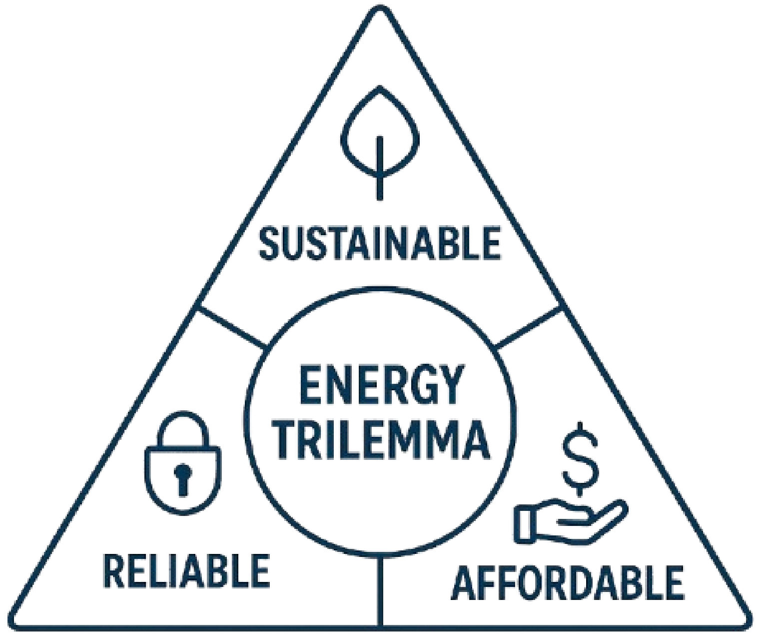
Policies & action. (n.d.). Climateactiontracker.Org. Retrieved 4 April 2025, from https://climateactiontracker.org/countries/kenya/policies-action/



Tackling the Challenges of Nigeria's Energy Transition Plan. (n.d.). Thisdaylive.Com. Retrieved 4 April 2025, from https://www.thisdaylive.com/index.php/2023/01/03/tackling-the-challenges-of-nigerias-energy-transition-plan/

The Energy Trilemma

- Sustainable - climate change. Decarbonisation
  → integrating RE, e.g., solar, wind
- Reliable - grid stability, balance demand and supply
  → VRE, flexible demand, infrastructure
- Affordable - investments and operational costs
  → Optimising investments and operations



Need for tools: complexity, optimisation, planning

Energy System Modelling (ESM)

Creating Digital Twins for Energy Systems

ESM is the process of creating simplified, computer-based representations (models) of energy systems.

Purpose

Understanding complexity, decision support and addressing the trilemma

Inputs

Technical data, economic data, demand data, resource data, policy data

Outputs

Optimal generation mix, investment decisions, operational schedules, energy flows, systems emissions, system costs, marginal prices

Why Python for ESM

EMS is data-heavy (time series, geospatial, economic)

Large ecosystem (libraries: Numpy, Pandas, PyPSA, etc.)

Open source and collaborative

“Glue language”: flexibility and integration.

Community support (e.g., Python Nairobi)

Rapid prototyping, readable syntax

NumPy pandas xarray matplotlib PyPSA pvlib

Calliope GeoPandas oemof ['ø:mɔf] open energy modelling framework PYOMO

Components

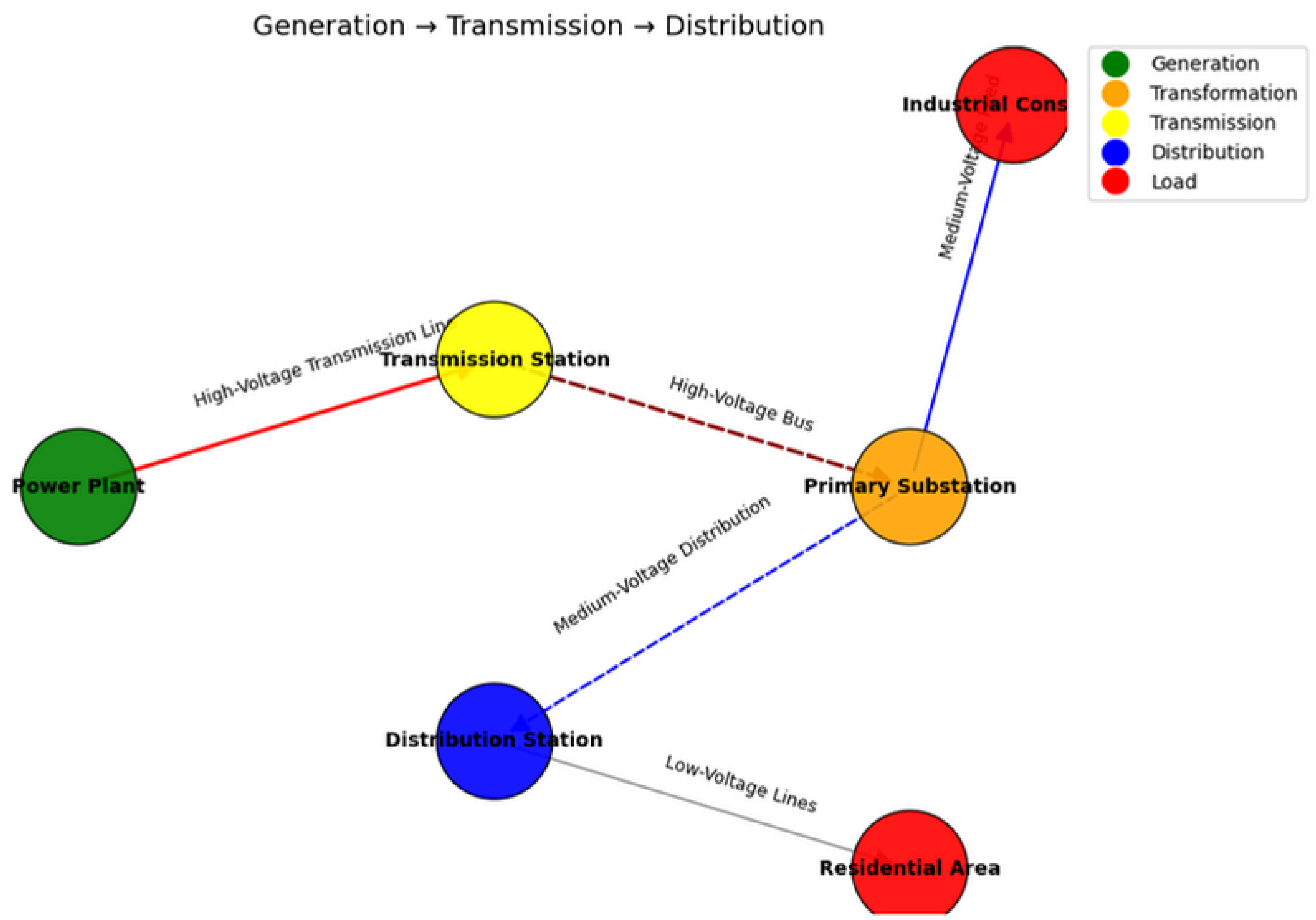
Energy Systems Refresher

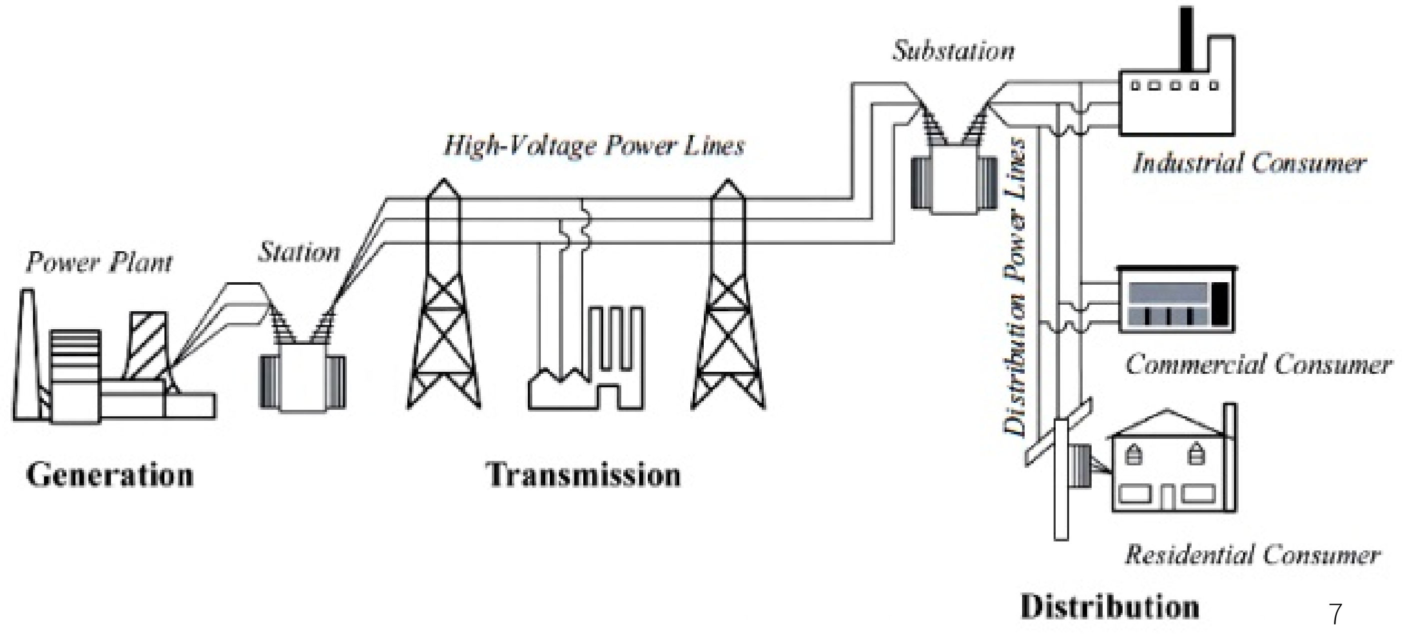
- **Generators**: Produce energy, e.g., solar, wind, gas, hydro.
  - Max capacity (MW), cost (to build & run), efficiency, availability (especially for renewables), emissions.

- **Loads**: Consume energy, e.g., cities, factories, residential areas.
  - Energy demand profile

- **Network**: Transports energy. The grid infrastructure. Consists of:
    - Buses (Nodes): Connection points where components meet (like electrical substations or junctions).
    - Lines & Transformers (Links): Carry power between buses.
  - Capacity, length, voltage, impedance.

- **Storage**: Shifts energy in time, e.g., batteries, pumped hydro storage.
  - Energy capacity (MWh), power capacity (MW for charging/discharging), efficiency.





BiNus Library | bab2BMC20150034

Physics of Flows

Energy Systems Refresher

- Kirchhoff's Voltage Law (KVL) $$\sum_{i=1}^{n} V_i = 0$$

Where:

- V_i = Voltage difference across the i^{th} component in a closed loop
- n = Number of components in the loop

- Kirchhoff's Current Law (KCL) $$\sum_{j \in neighbors(i)} P_{ij} = 0$$

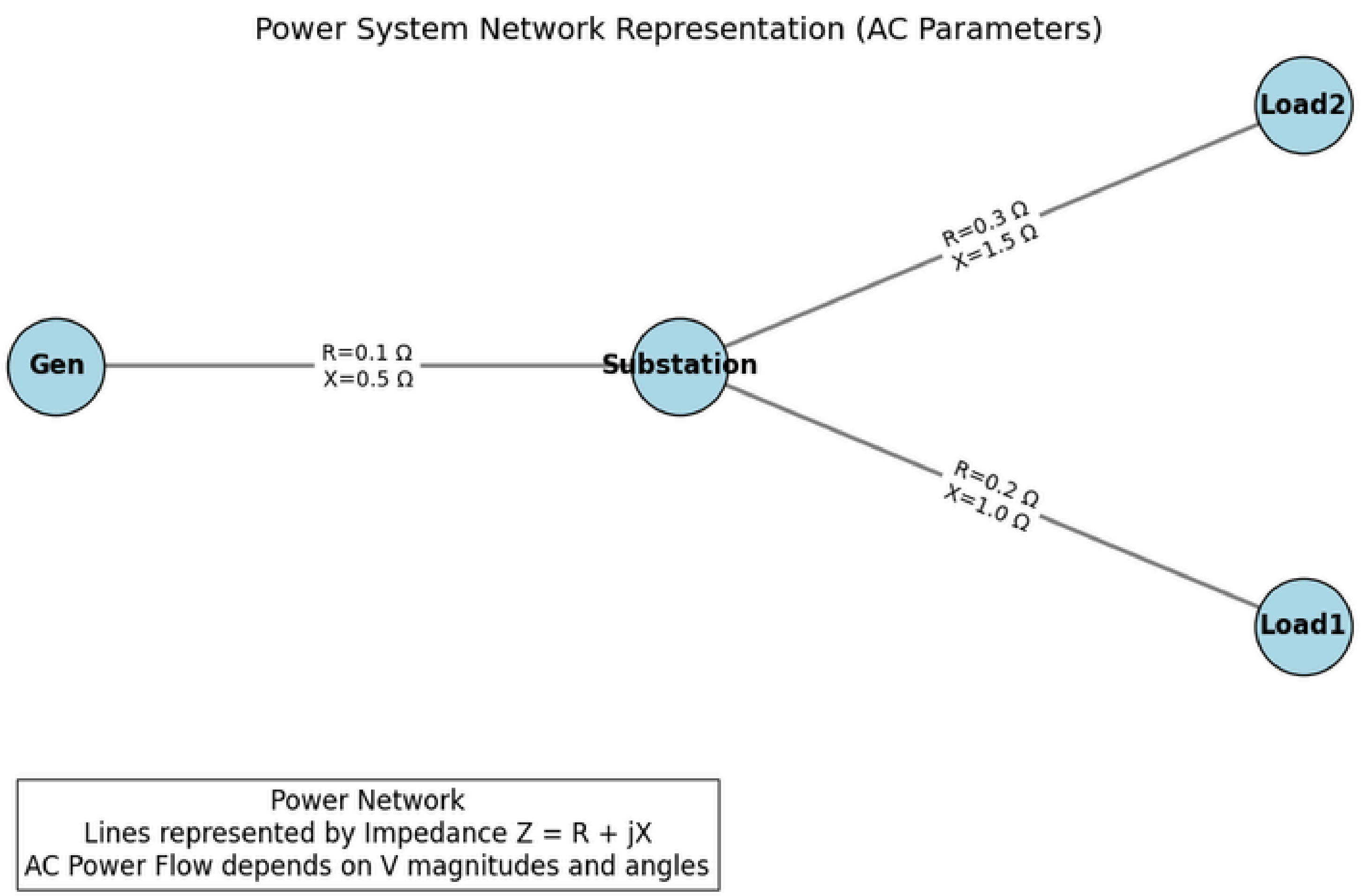
Where:

- P_{ij} = Power flow from node i to node j
- The sum is taken over all neighbouring nodes j connected to node i

- Ohm's Law $$V = I \; x \; R$$

Where:

- V = Voltage (Volts)
- I = Current (Amperes)
- R = Resistance (Ohm)

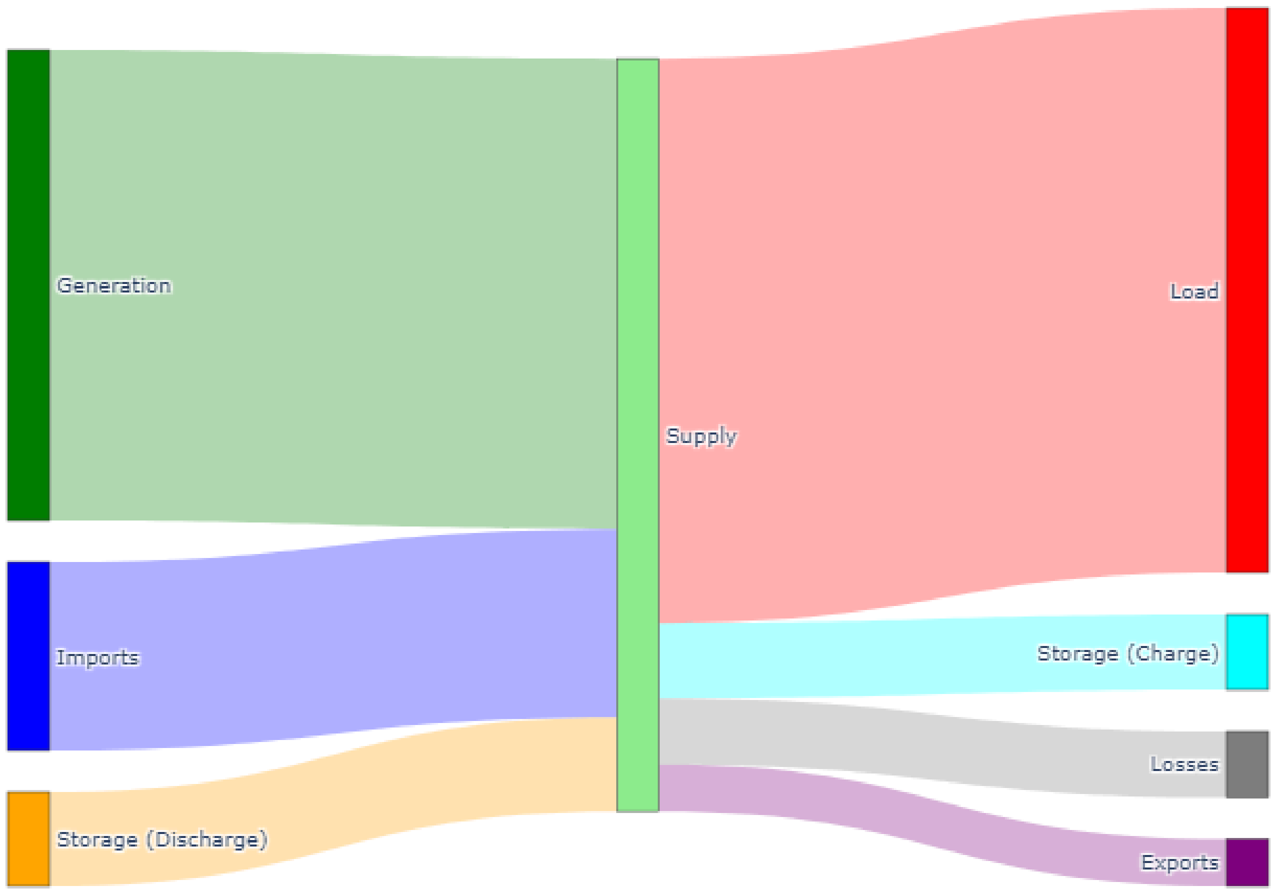


Energy Balance

Energy Systems Refresher

- At every bus and at every moment in time: **Energy In = Energy Out**.

Generation + Imports + Storage Discharge = Load + Exports + Storage Charging + Losses



For each time t, a demand of d_t and a *'per unit'* availability w_t for wind and s_t for solar, we have $W\ MW$ of wind and $S\ MW$, the effective residual load or mismatch is

$$m_t = d_t - Ww_t - Ss_t$$

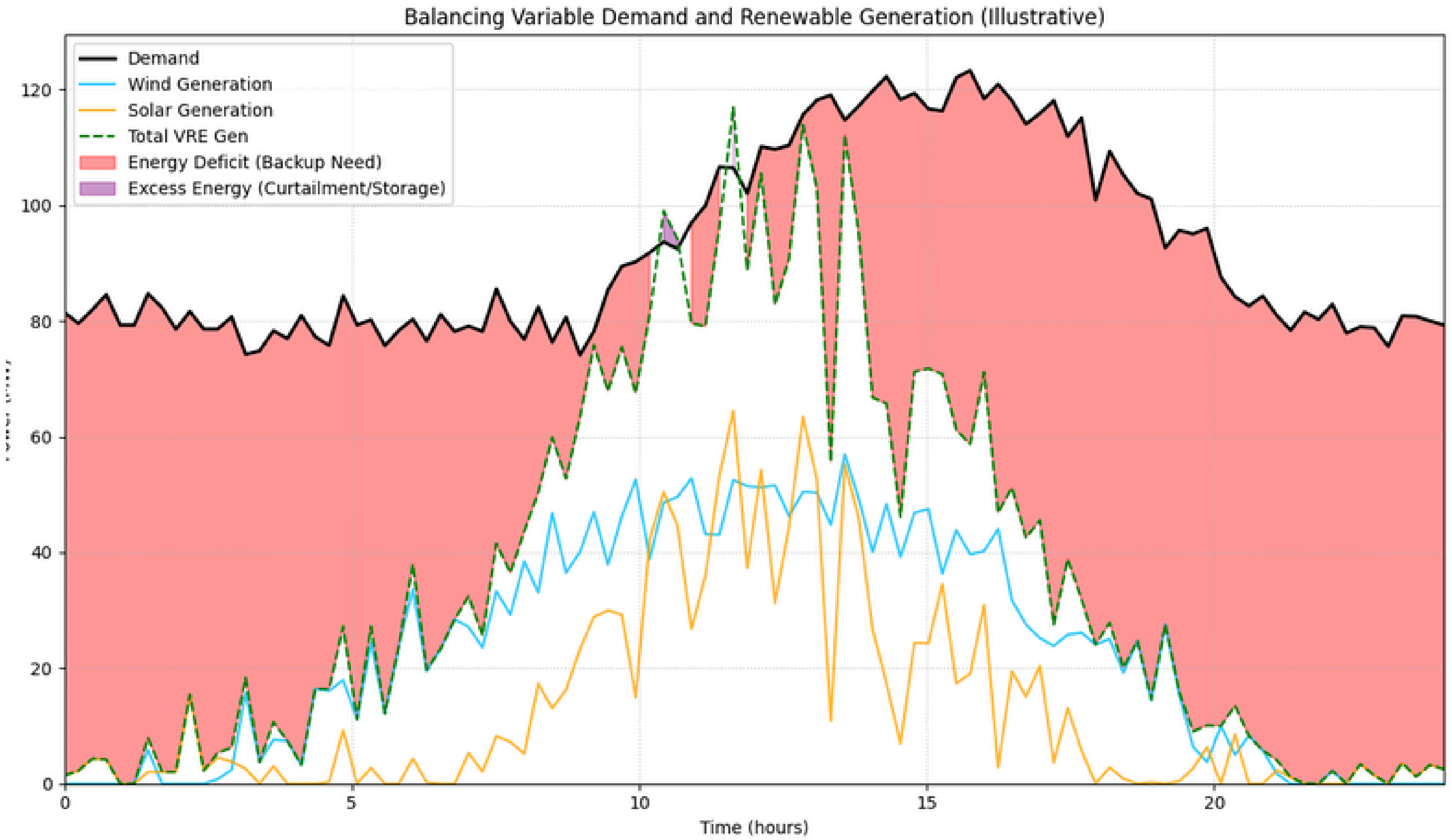
Let p_t be the balance of power at all times. $p_t = 0$ because we cannot create or destroy energy.

If $m_t > 0$, we need backup power $b_t = m_t$ to cover the loads so that

$$p_t = b_t - m_t = b_t - d_t + Ww_t + Ss_t = 0$$

If $m_t < 0$, we need curtailment $c_t = -m_t$ to reduce the excess feed-in so that

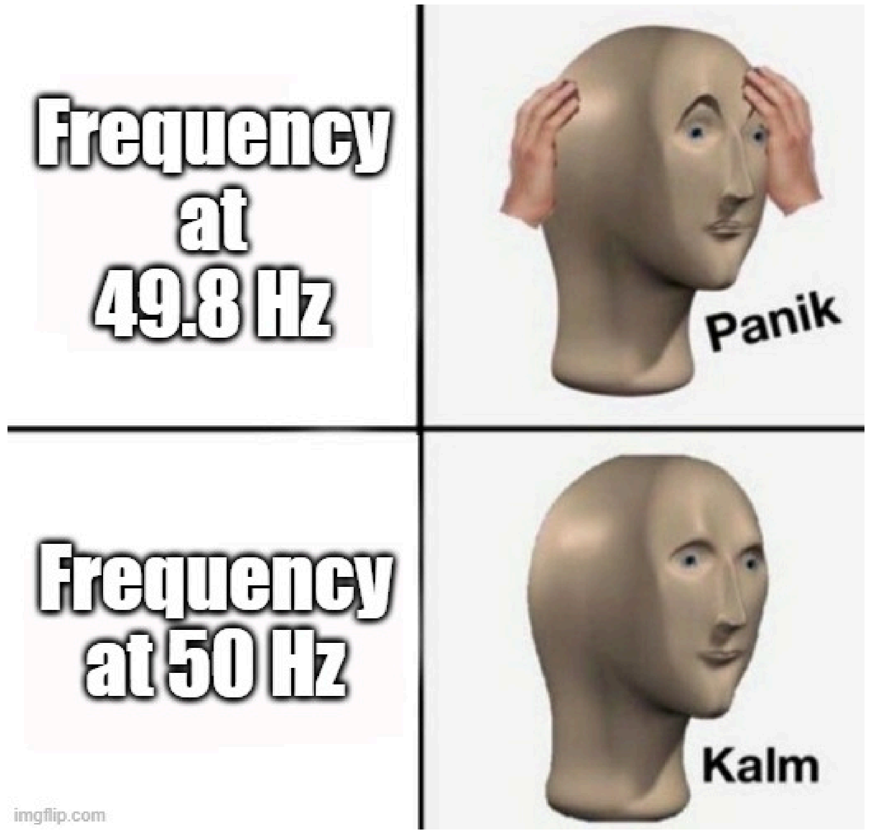
$$p_t = b_t - c_t = c_t - d_t + Ww_t + Ss_t = 0$$

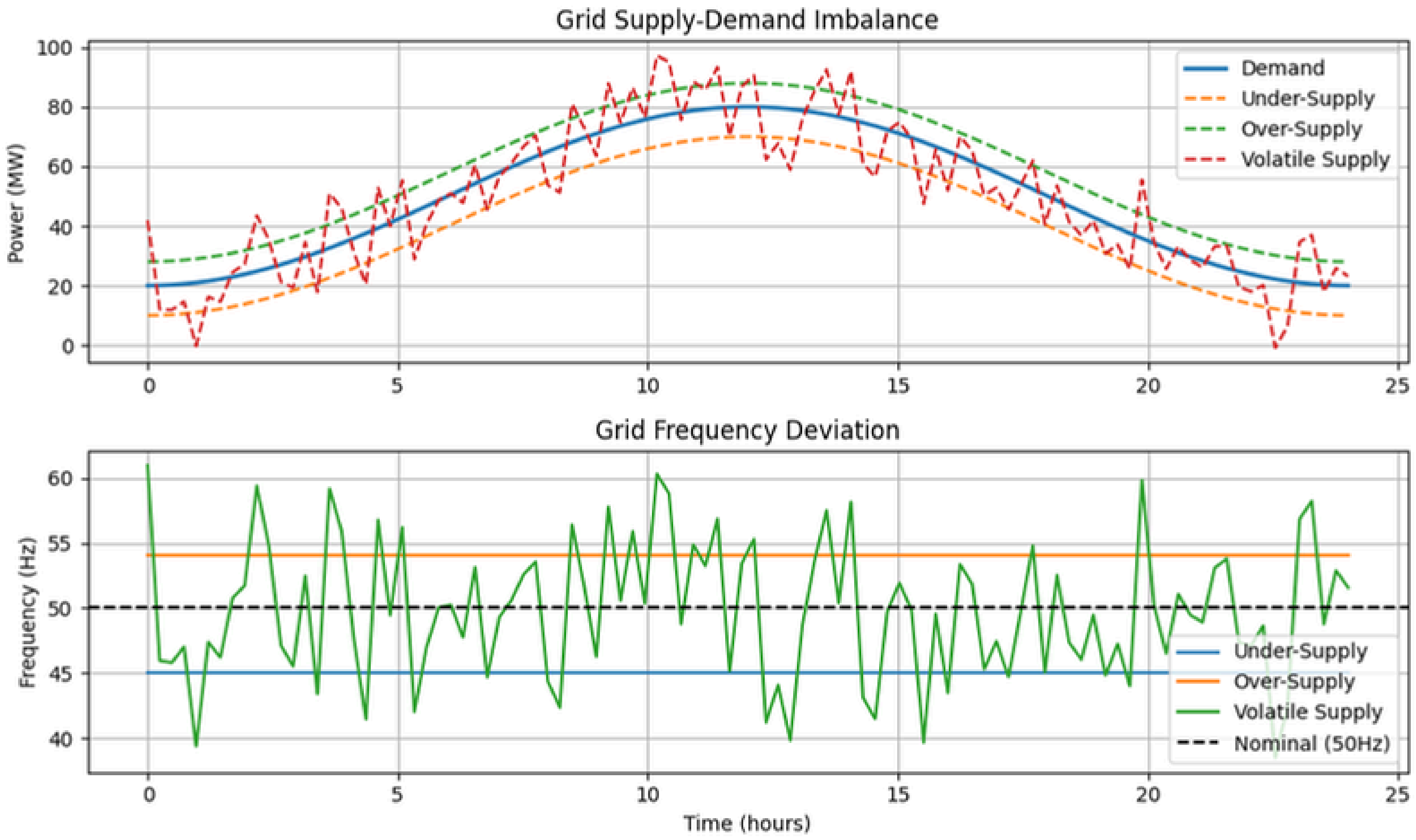


Energy Balance

Energy Systems Refresher

- If supply ≠ demand => frequency deviations
- $\Delta f \propto \Delta P / H$

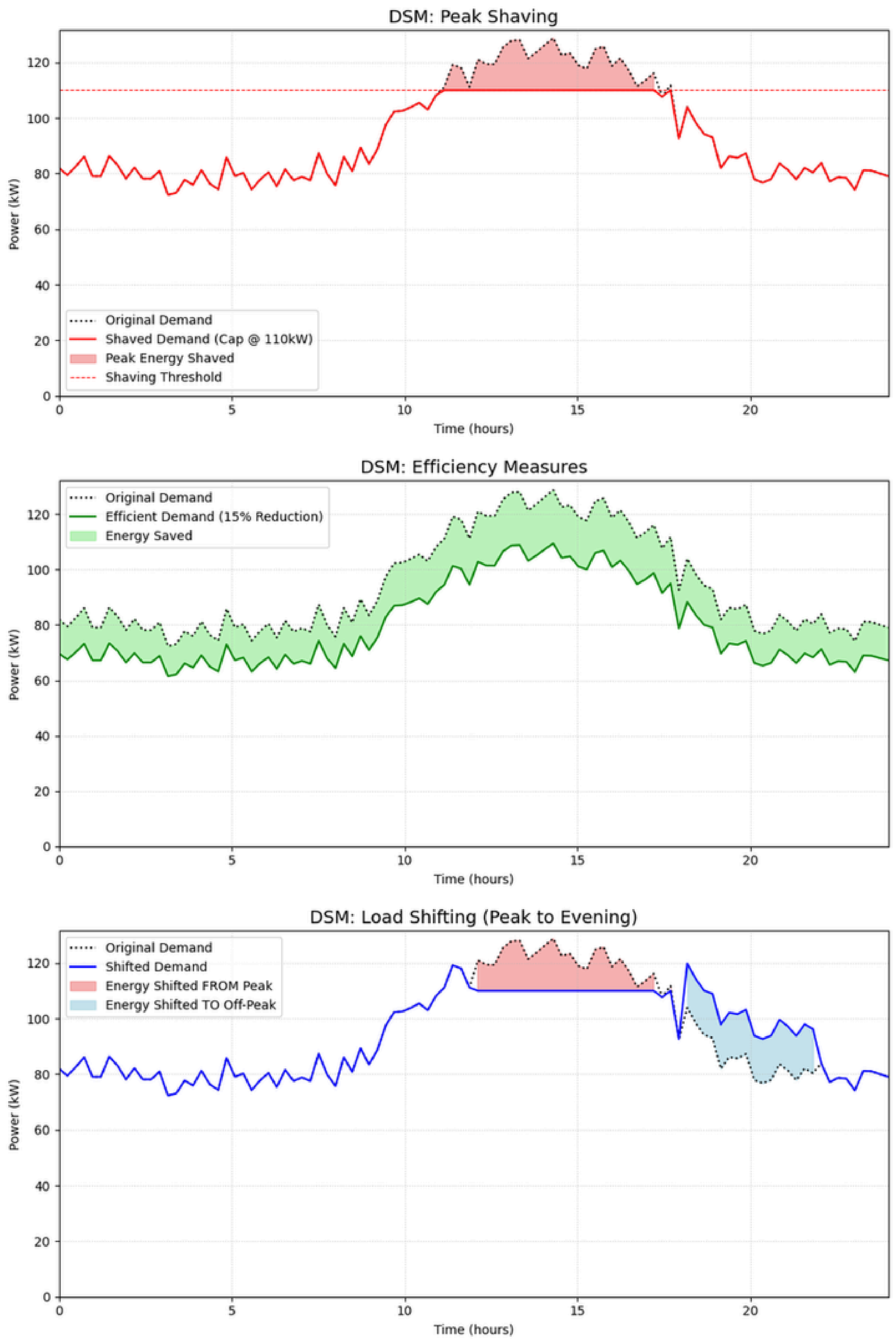




Dynamics & Variability

Energy Systems Refresher

- **Demand-Side Management (DSM):** aims to optimise energy usage by adjusting the demand rather than increasing supply.
    - **Load Shifting:** Moving energy use from peak times to off-peak periods.
    - **Efficiency Measures:** Reducing overall energy consumption without impacting performance.
    - **Peak Shaving:** Minimising the highest peaks in energy demand.



Images for illustration generated by a Python script

Time is Crucial

Energy Systems Refresher

- **DSM vs. Storage:**
    - Storage shifts energy availability over time, DSM directly manages demand patterns to optimize energy usage

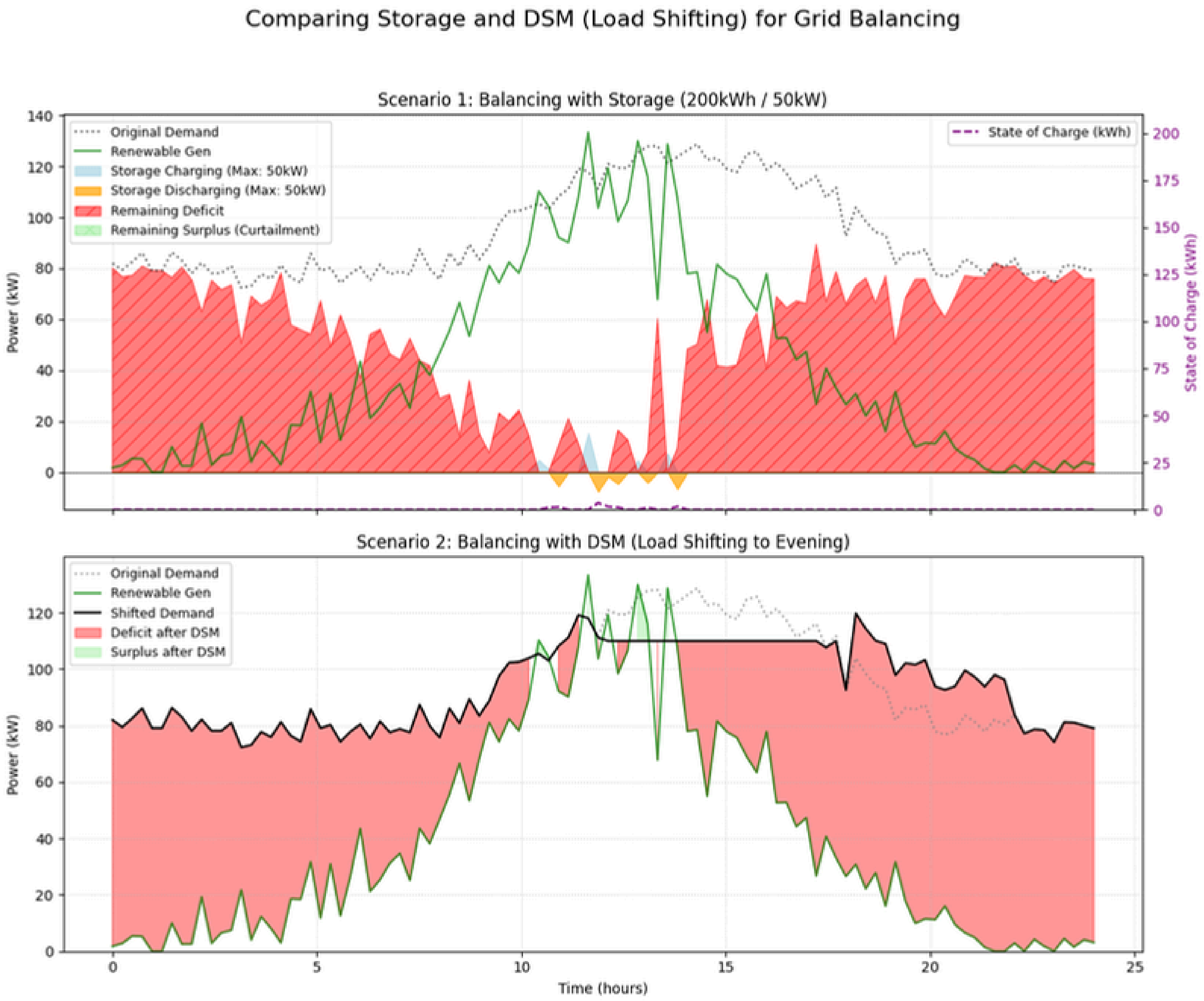


Image for illustration generated by a Python script

Energy Economics

Energy Systems Refresher

Cost of Energy Production:
- Factors: Fuel costs, CAPEX, OPEX
- Levelised Cost of Energy (LCOE)

Energy Pricing:
- Influences: market structure, supply-demand dynamics, policies. Peak vs. off-peak pricing (time-of-use tariffs).

Economic Dispatch:
- Optimising the generation mix to minimise costs while meeting demand.

Market Mechanisms:
- Spot pricing, power purchase agreements (PPAs), and subsidies.

Cost-optimal energy system

$$minimise\ (yearly\ system\ costs) = \sum_{n}(Annualised\ capital\ costs) + \sum_{n,t}(marginal\ costs)$$

Subject to: meeting demand at each node, n at time, t; availability time series; transmission constraints; capacity $\leq$ geographical potentials; and emissions (CO_2)

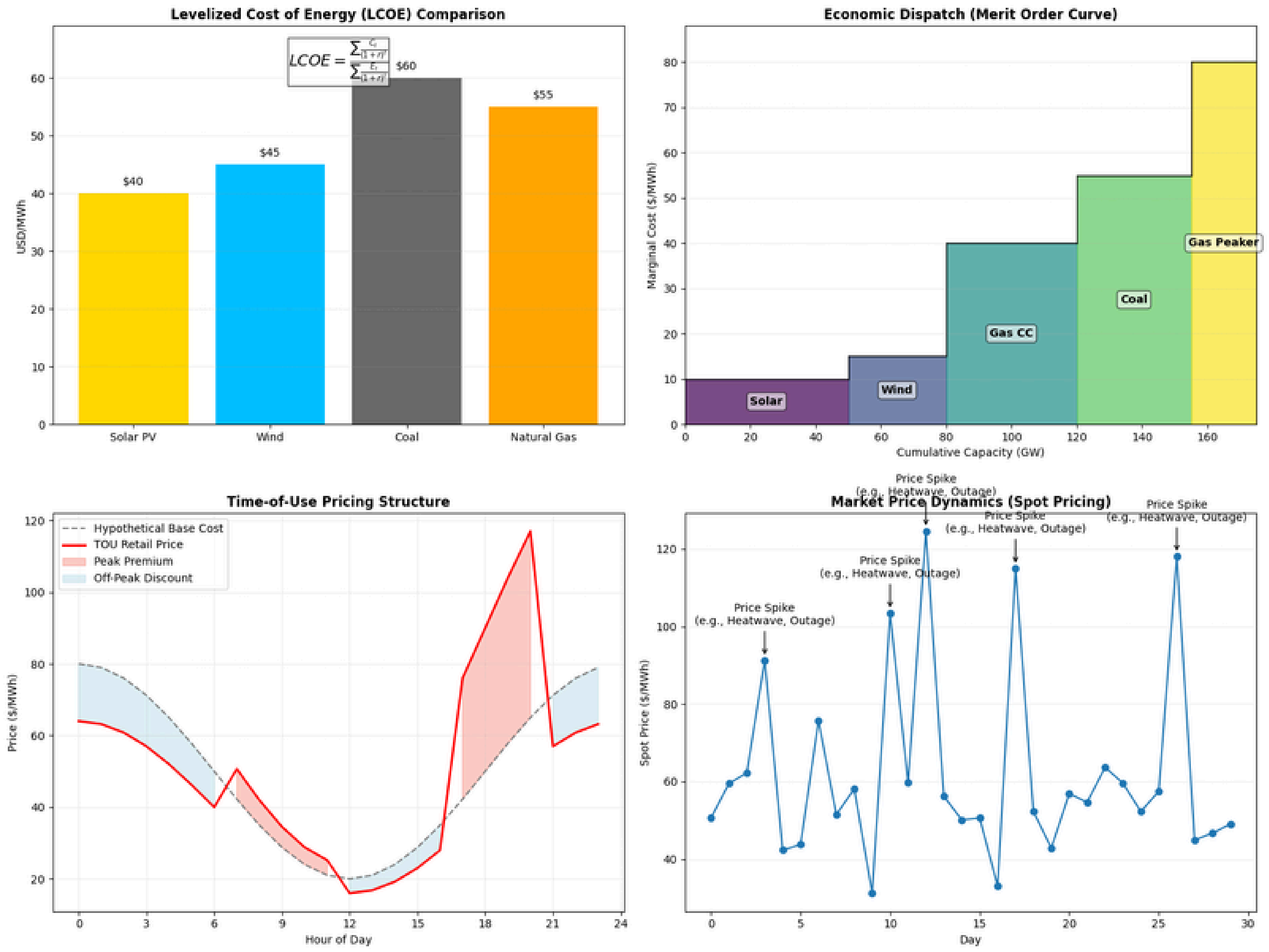


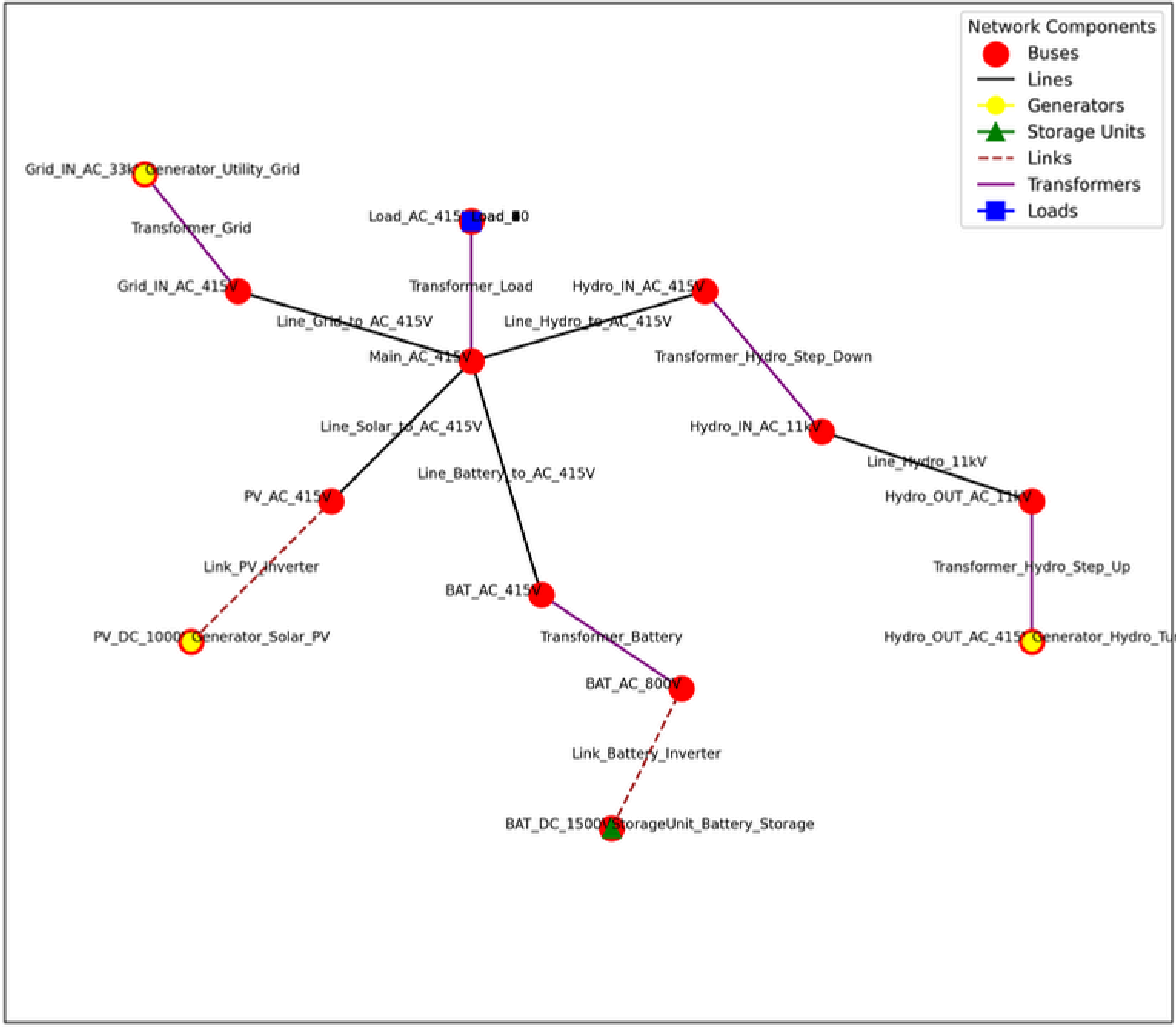
Image for illustration generated using a Python script

ESM with PyPSA Step-by-Step

Jupyter Notebook Tutorials

https://github.com/tinegachris/ESM-Python-Nairobi/tree/main/notebooks/tutorials

- **01_getting_started.ipynb**: Introduces PyPSA. Covers basic network creation, adding time series and basic components.
- **02_components_basics.ipynb**: Covers the basic components available in PyPSA and how to configure them.
- **03_network_visualization.ipynb**: Covers how to visualise PyPSA networks using various plotting functions.
- **04_basic_optimization.ipynb**: Covers basic network optimisation in PyPSA.
- **05_investment_planning.ipynb**: Covers how to perform investment planning in PyPSA.
- **06_storage_and_balancing.ipynb**: Covers how to model storage systems and demand-side management in PyPSA.
- **07_sector_coupling.ipynb**: Covers how to model sector coupling in PyPSA and their interactions using coupling technologies (e.g., heat pumps, electrolyzers, and fuel cells).
- **08_diy_solar_simulation.ipynb**: Demonstrates how PyPSA can model a small, real-life energy system, like a Solar Home System (SHS).



Network Components
Buses
Lines
Generators
Storage Units
Links
Transformers
Loads
Grid_IN_AC_33k
enerator_Utility_Grid
Transformer_Grid
Grid_IN_AC_415V
Load_AC_415
Transformer_Load
Line_Grid_to_AC_415V
Main_AC_415V
Hydro_IN_AC_415V
Line_Hydro_to_AC_415V
Transformer_Hydro_Step_Down
Line_Solar_to_AC_415V
Hydro_IN_AC_11kV
Line_Hydro_11kV
Line_Battery_to_AC_415V
PV_AC_415V
Hydro_OUT_AC_11kV
Link_PV_Inverter
Transformer_Hydro_Step_Up
BAT_AC_415V
PV_DC_1000
enerator_Solar_PV
Transformer_Battery
Hydro_OUT_AC_415
enerator_Hydro_Turbine
BAT_AC_800V
Link_Battery_Inverter
BAT_DC_1500V
torageUnit_Battery_Storage

tinegachris/
hybrid_renewable_pypsa

Analysis of a hybrid renewable energy network using PyPSA (Python for Power System Analysis)

1 Contributor | 0 Issues | 0 Stars | 0 Forks

GitHub - tinegachris/hybrid_renewable_pypsa: Analysis of a hybrid renewable energy network using PyPSA (Python for Power System Analysis)

Link https://github.com/tinegachris/hybrid_renewable_pypsa

Practical Tips on Effective ESM

- GIGO
- Define your question first to define scope, complexity and resolution
- Start simple; add complexity gradually
- Understand model assumptions and limitations. How do they affect results?
- Validate and sanity check
- Leverage community resources
- Choose the right tools

Python for Energy System Modelling

Python-Nairobi

Chrispine Tinega

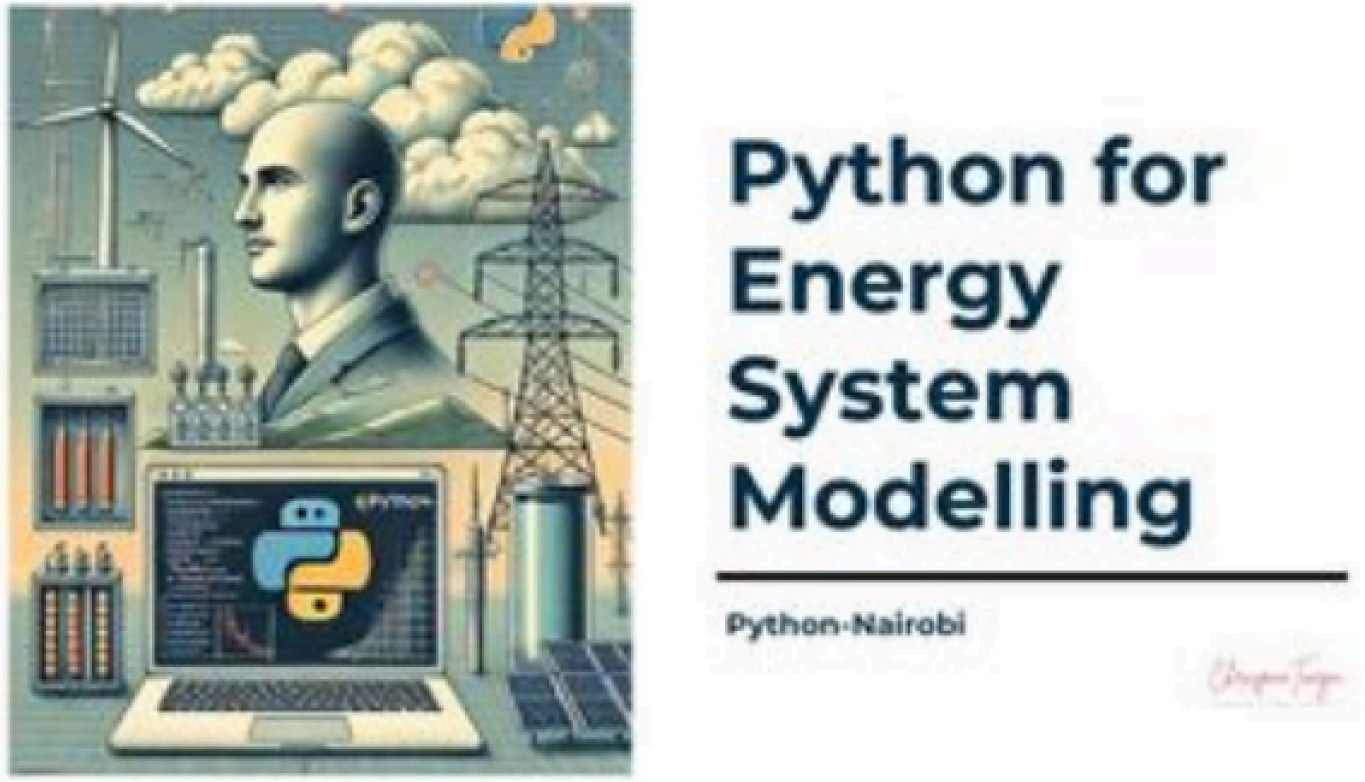
GitHub - tinegachris/ESM-Python-Nairobi: Documentation of Python for Energy System Modelling event hosted by Python Nairobi User Group.

Link https://github.com/tinegachris/ESM-Python-Nairobi

Continue Your Journey ♥

- PyPSA Documentation: The official guide, tutorials, examples, and API reference – https://pypsa.readthedocs.io/
- Tom Brown's Energy Systems Modelling Course: Comprehensive university course materials covering PyPSA and general concepts – https://nworbmot.org/courses/
- Fabian Neumann's Data Science for ESM Course: Focuses on data handling, analysis, and visualization in energy system modelling – https://fneum.github.io/data-science-for-esm/
- Open Energy Modelling Wiki (Learning Materials): A curated list of diverse learning resources across the open energy modelling ecosystem – https://wiki.openmod-initiative.org/wiki/Learning_materials
- Atlite Documentation: Crucial tool for converting weather data into time series for renewables and demand – https://atlite.readthedocs.io/
- Linopy Documentation: The optimization modelling framework underpinning recent versions of PyPSA – https://linopy.readthedocs.io/
- PyPSA-Eur: A large-scale, open model of the European transmission system – https://github.com/PyPSA/pypsa-eur
- PyPSA-Africa: An open model focusing on the African continent's power system – https://github.com/PyPSA/pypsa-africa
- OSeMOSYS (Main Website): A widely used framework for long-term energy system planning and integrated assessment – http://www.osemosys.org/
- Starter Kits (Examples): Practical examples to kickstart energy modelling – https://climatecompatiblegrowth.com/starter-kits/
- Open University Course: Online course to deepen your energy systems knowledge – https://www.open.edu/openlearncreate/course/view.php?id=6817
- Open Energy Modelling Initiative Forum: A platform for discussions on all aspects of open energy modelling – https://forum.openmod-initiative.org/
- Python Documentation: Official reference for the Python language – https://docs.python.org/3/
- Google OR-Tools: Introduction to the optimization solvers these frameworks rely on – https://developers.google.com/optimization/introduction

Open Session



Python for Energy System Modelling, Sat, Apr 5, 2025, 10:00 AM

Join us for an exciting evening of learning and networking with fellow Python enthusiasts at the Python Nairobi User Group. Our guest speaker, Chrispine Tinega, will be del

Meetup / Apr 5

Thank You!