ModuGen Backend Coding Challenge

Welcome to the ModuGen backend coding challenge.

Task

- We want you to build a REST API in Python that allows us to cut polygons at planes in 3D space
    - For the sake of simplicity, we assume that all polygons are defined in the XY plane, but the user might not know this
        - the data representation should allow for the API to be extended to full 3D interactions later on.
- The api takes the polygon and the plane and returns the cut result
- You can assume that the polygon must be convex, but you cannot assume the user to know this
- You can assume that the plane is always orthogonal to the polygon, but you cannot assume the user to know this

Evaluation

- We check out the repo and launch the api (on a different machine than the dev machine ⇒ make sure setup is reproducible)
- We try some requests and discuss the implementation and possible extensions / improvements.
- We evaluate code-style, best-practices for production ready code (environment, docs, testing, …) and your ability to complete this task with minimal instructions.

Hints

- For the given limitations, it makes sense to simplify the cut implementation to 2D space
- If needed, you can add additional constraints on the geometries (like maximum size, etc…) if it simplifies the implementation.
- You can choose the tooling you use freely.

Evaluation

- Challenges:
    - Choose polygon and plane representations appropriately, both for api user and for calculations
    - Ensure appropriate data validation and error handling

- Expectations
    - Edge case handling: no cut, single cut result (edge), 2 cut results
    - Order of cut results is consistent and predictable in some way
    - Polygons with points that are not on the same plane raise an error
    - Correct docker configuration for api setup
    - Dependency management
    - Slides for explanation
    - Testing
    - Layering separation of concerns

- What we want to see
    - Clear separation of concerns
    - Layering
    - clean code
        - function naming
        - formatting
        - …
    - environment setup
    - docker setup to run the whole application
    - commit history / working style
    - testing