

# Design-Focused Questions: Power Grid Simulation System

## Question a) Can a power company own more than one plant?

\*\*Yes.\*\*

The relationship between `PowerCompany` and `PowerPlant` is defined as a one-to-many composition with cardinality `1` to `1..\*`. This means a single power company must own at least one power plant and can own multiple plants. The `PowerCompany` class includes an `addPowerPlant(plant: PowerPlant)` operation to support adding additional plants, and the bidirectional navigation `company.getPowerPlants()` returns a collection.

---

## Question b) Can the rate charged by a company be changed over time?

\*\*Yes.\*\*

The `ratePerKWh` attribute in the `PowerCompany` class is a mutable decimal value, not a constant. This design allows the company to update its standard rate at any time. Since customer bills are calculated dynamically using `calculateBill(rate: Decimal)` at billing time, any rate change will automatically apply to subsequent billing cycles.

---

## Question c) Can residential and commercial customers be charged at different rates?

\*\*No, not in the current design.\*\*

The `PowerCompany` class has a single `ratePerKWh` attribute that applies uniformly to all customers. While `Customer` has a `customerType` attribute (RESIDENTIAL or COMMERCIAL) for classification

purposes, there is no mechanism to associate different rates with different customer types. To support differentiated rates, the design would need either rate attributes per customer type in `PowerCompany` or a rate attribute in `Customer`.

---

#### **Question d) Can separate customers of the same type be charged at different rates?**

**\*\*No.\*\***

Following from the answer to question (c), all customers are charged at the company's uniform `ratePerKWh`. Individual customers do not have their own rate attribute. The `Customer.calculateBill(rate: Decimal)` method receives the rate as a parameter from the company, and all customers of the same company receive the same rate regardless of their type or individual circumstances.

---

#### **Question e) Can a substation or transformer change the source of its power after creation?**

**\*\*Yes.\*\***

The relationships `PowerPlant` → `Substation` and `Substation` → `Transformer` are modeled as aggregations (not compositions), indicating that the child components have a lifecycle independent of their parent. The `PowerPlant` class includes both `addSubstation()` and `removeSubstation()` operations, and `Substation` includes `addTransformer()` and `removeTransformer()` operations. This allows a substation to be disconnected from one plant and reconnected to another, and similarly for transformers.

---

#### **Question f) If the power source for a substation or transformer is changed, can the new source be owned/managed by a different company?**

**\*\*No.\*\***

The referential integrity constraints (R2, R3, R4) establish that `Transformer` → `Substation` → `PowerPlant` → `PowerCompany` forms a strict ownership chain. Since `Customer` is associated with a `PowerCompany` through composition, and customers connect to transformers, allowing cross-company connections would break the billing and service relationship. Additionally, `Employee` dispatch for `EquipmentIssue` is scoped to a single company. Therefore, all equipment in a distribution chain must belong to the same company.

---

### **Question g) Can the capacity of an element be changed over time?**

**\*\*No.\*\***

The capacity limits (20 substations per plant, 10 transformers per substation, 5 customers per transformer) are defined as system constants (`MAX\_SUBSTATIONS\_PER\_PLANT`, `MAX\_TRANSFORMERS\_PER\_SUBSTATION`, `MAX\_CUSTOMERS\_PER\_TRANSFORMER`). These are not instance attributes but fixed values enforced by operations like `canAddSubstation()`, `canAddTransformer()`, and `canAddCustomer()`. To support variable capacities, each entity would need a `maxCapacity` attribute.

---

### **Question h) Is there a distance limitation/restriction between plants and substations, substations and transformers, and/or transformers and customers?**

**\*\*Yes.\*\***

- **\*\*D1\*\*:** Substation must be within `MAX\_PLANT\_TO\_SUBSTATION\_DISTANCE` from its PowerPlant
- **\*\*D2\*\*:** Transformer must be within `MAX\_SUBSTATION\_TO\_TRANSFORMER\_DISTANCE` from its Substation
- **\*\*D3\*\*:** Customer must be within `MAX\_TRANSFORMER\_TO\_CUSTOMER\_DISTANCE` from their Transformer

The `Location` value object provides the `manhattanDistance(other: Location)` and `isWithinDistance(other: Location, maxDist: Integer)` operations to validate these constraints. These distance checks should be performed in the `addSubstation()`, `addTransformer()`, and `addCustomer()` operations before establishing connections.