

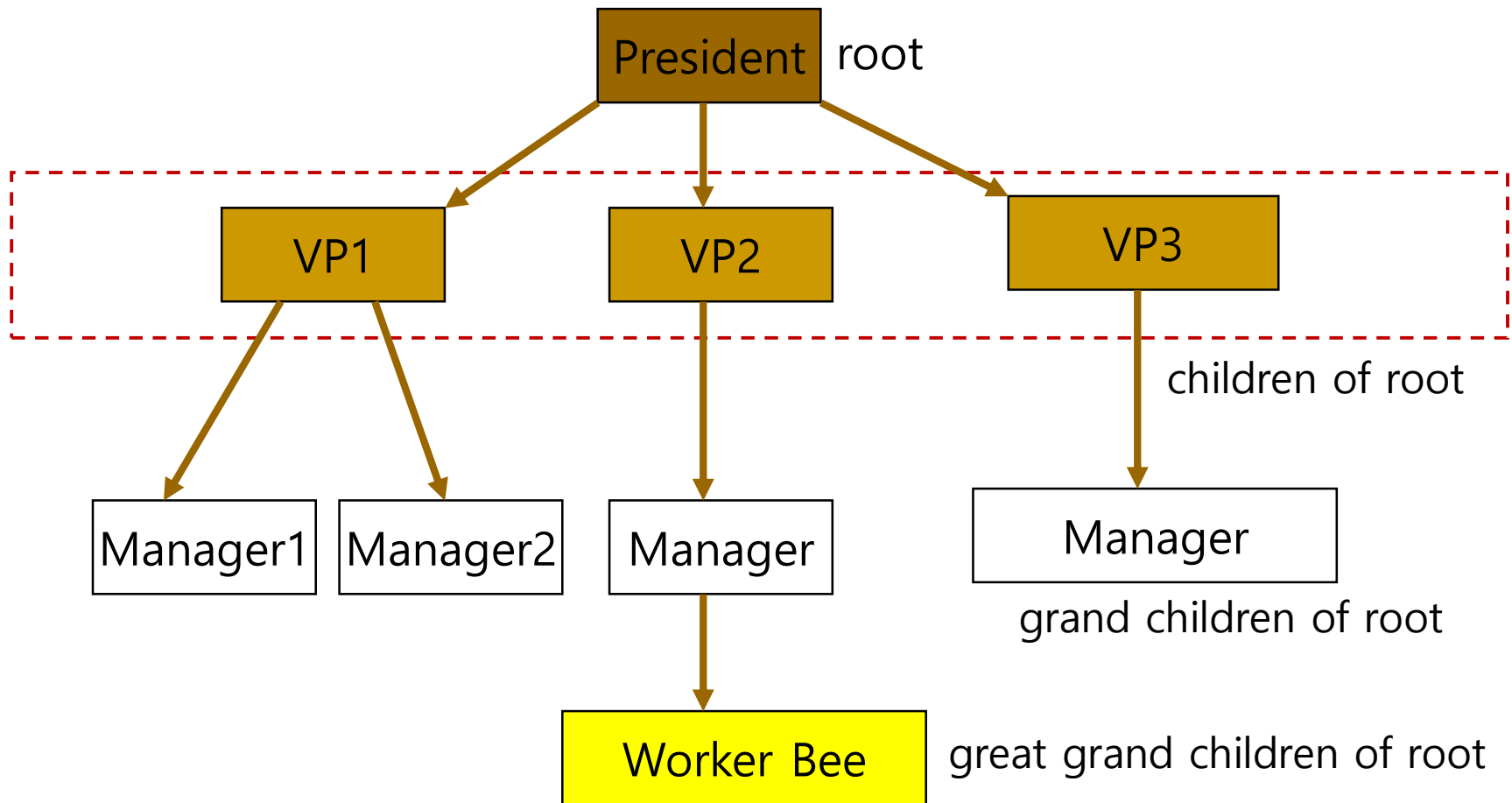


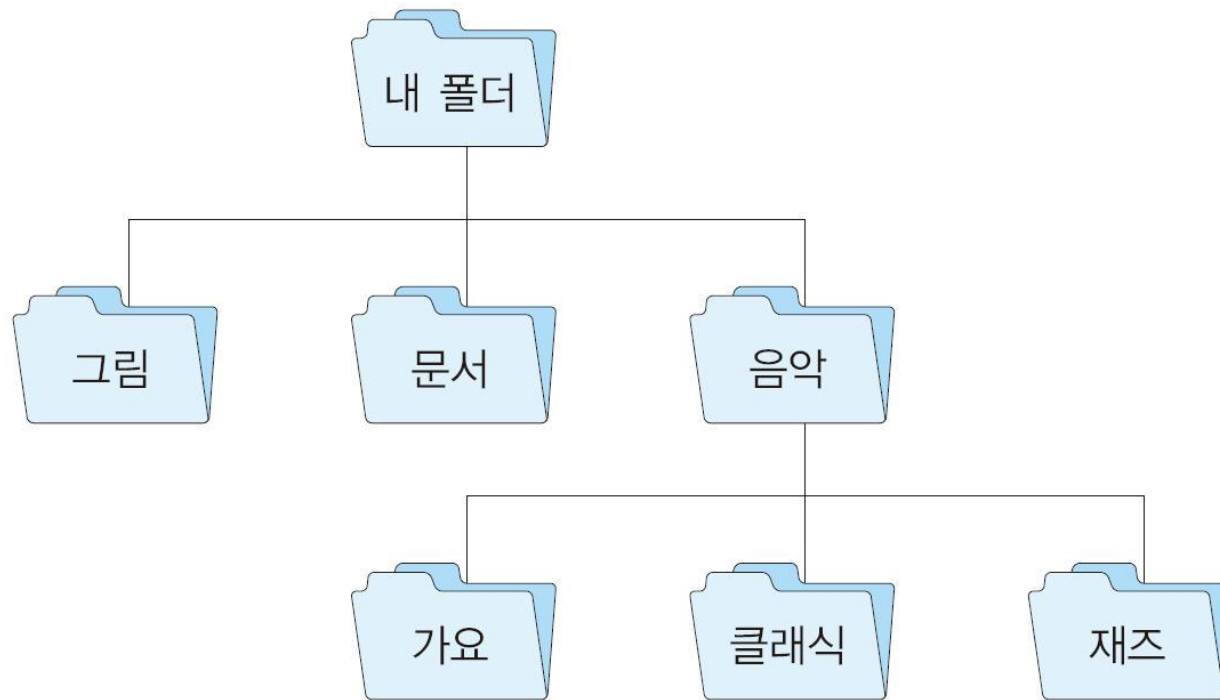
Tree

Linear List 와 Tree

- **Linear lists** are useful for serially ordered data.
 - $(e_0, e_1, e_2, \dots, e_{n-1})$
 - 순서를 가지면서 선형으로 배열된 데이터 처리
 - Days of week.
 - Months in a year.
 - Students in this class.
 - **Trees** are useful for hierarchically ordered data.
 - 상하 계층을 이루면서 수직적으로 표현되는 데이터 처리
 - Employees of a corporation.
 - President, vice presidents, managers, and so on.
 - UNIX 파일 시스템
-

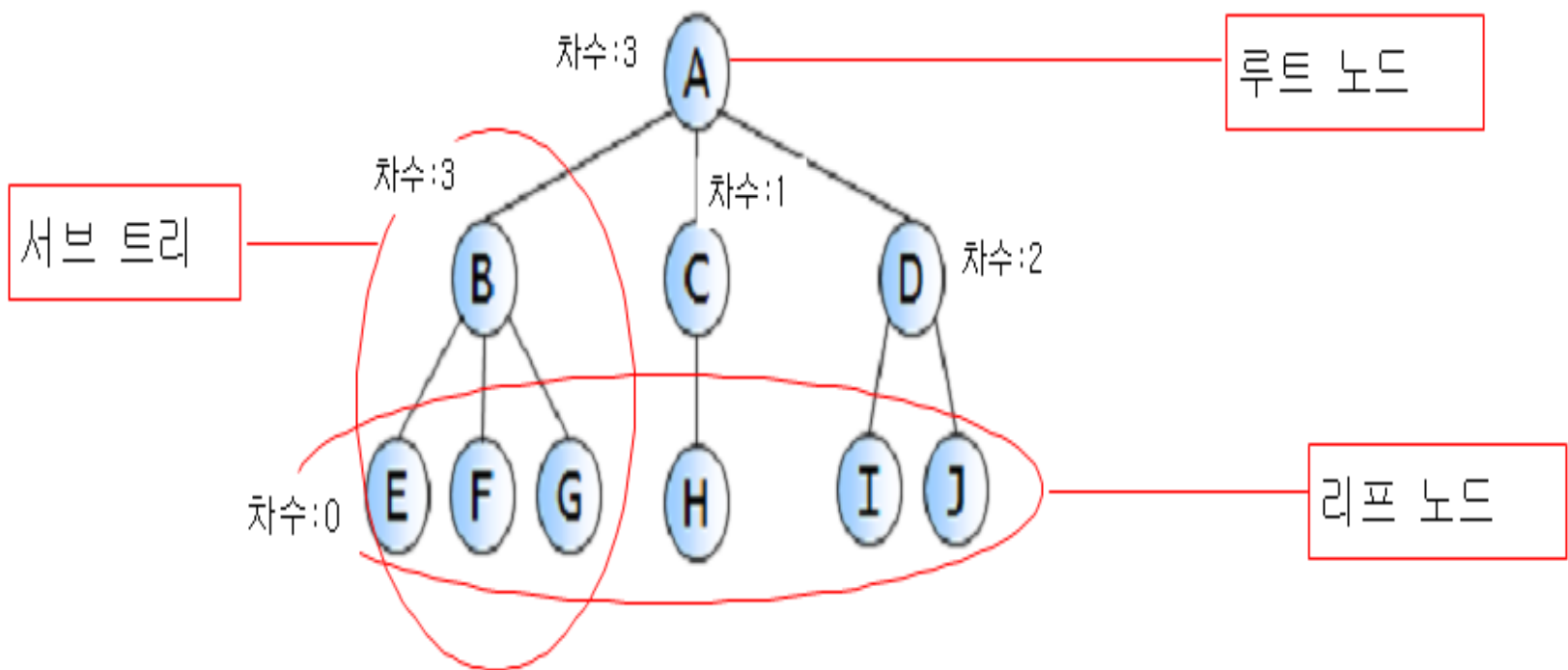
Example Tree





Hierarchical Data 와 Tree

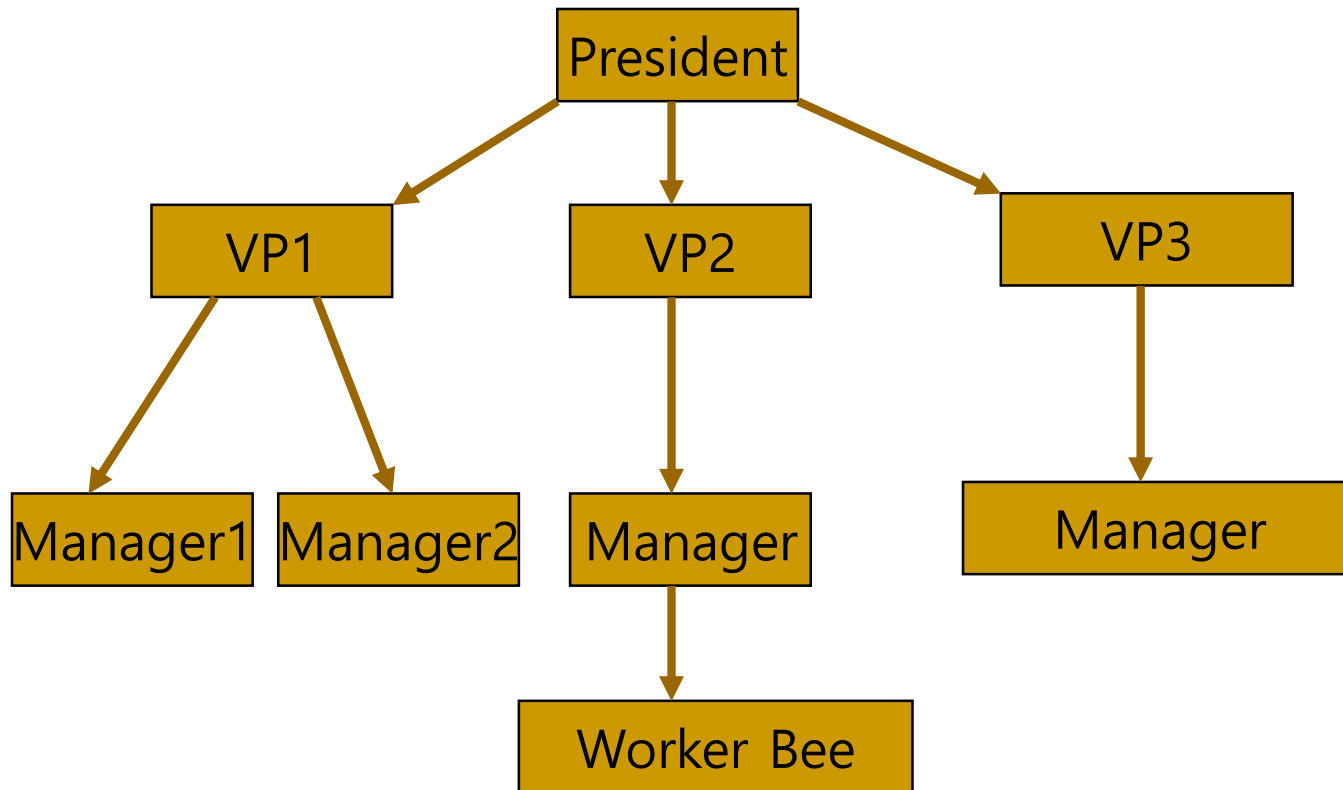
- **노드(node)** : 트리를 구성하는 각 vertex
- 계층의 top에 있는 노드를 root 라고 한다
- **자식(children) 노드** : 어떤 노드의 바로 하위의 모든 노드
- **부모(parent) 노드** : 어떤 노드의 바로 상위의 노드
- **형제(sibling) 노드** : 같은 부모 아래의 모든 자식 노드
- **조상(ancestor) 노드** : 어떤 노드의 상위에 있는 모든 노드
- **후손(descendant) 노드** : 어떤 노드의 하위에 있는 모든 노드
- **리프(leaf)노드** : 자식이 없는 노드
- **서브 트리** : 주어진 트리의 부분집합을 이루는 트리



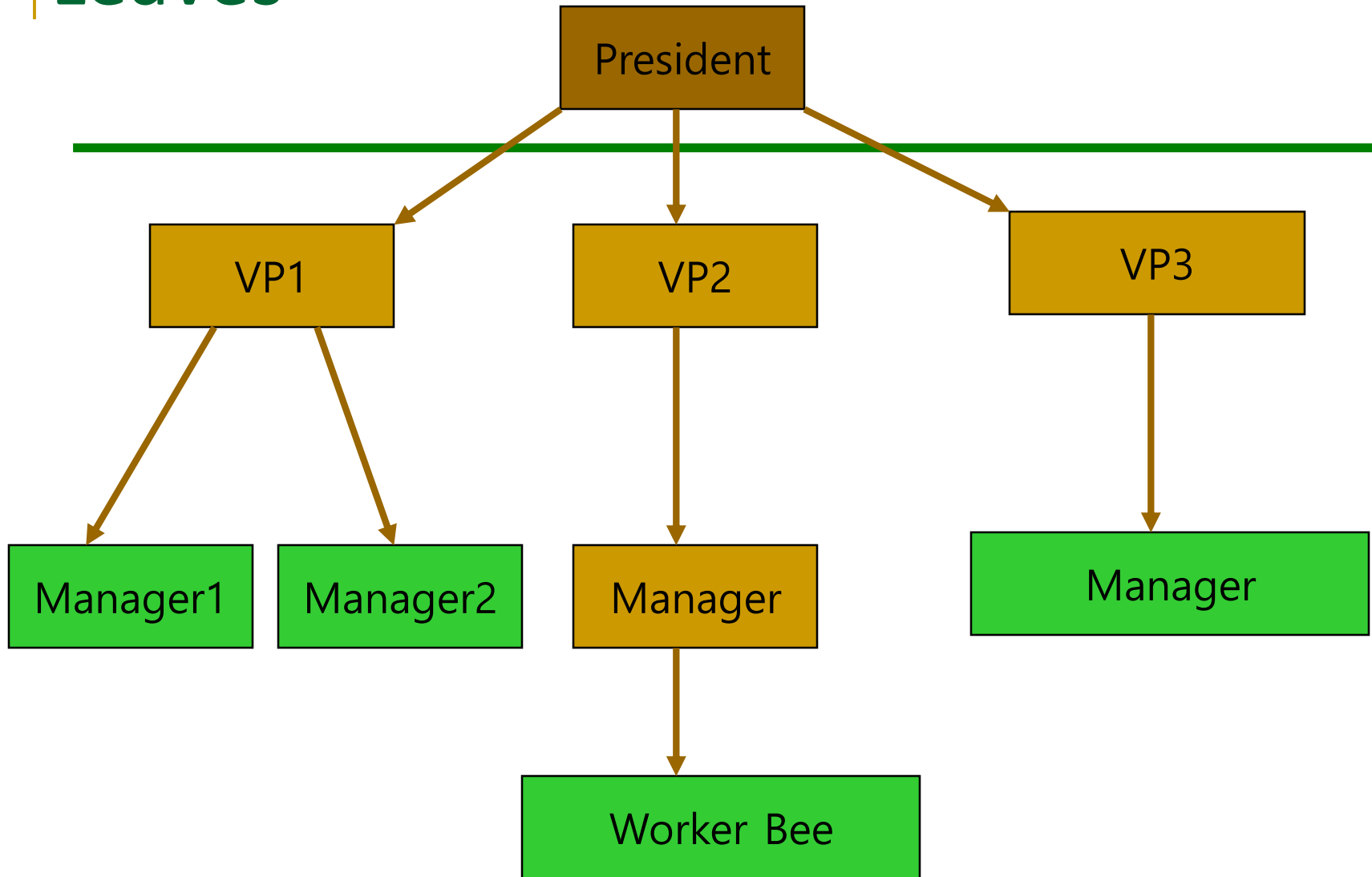
Tree의 성립 요건

-
- A tree T is a **finite nonempty set** of elements.
 - One of these elements is called the **root**.
 - The remaining elements, if any, are partitioned into trees, which are called the **subtrees** of T
-

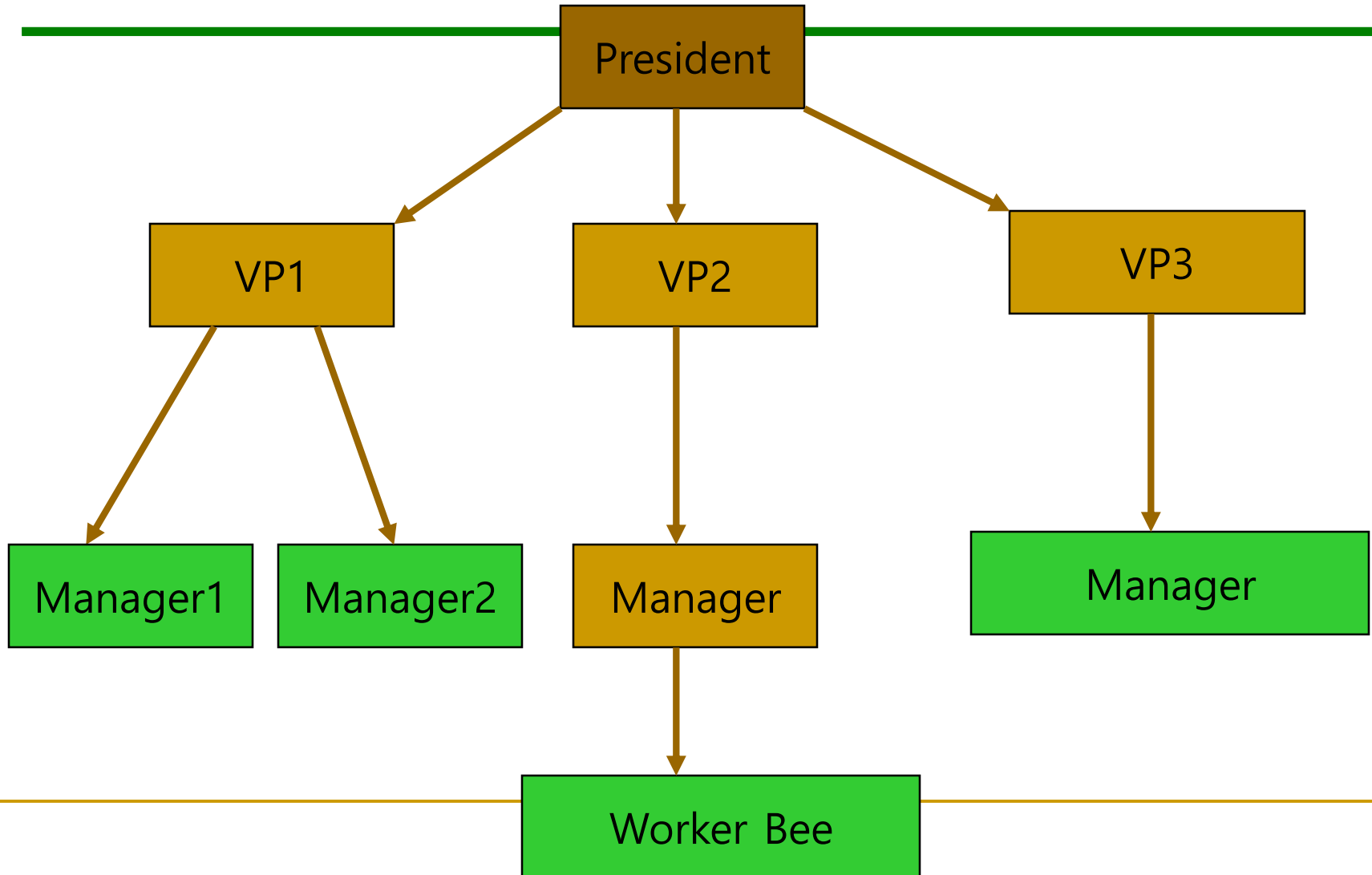
Subtrees



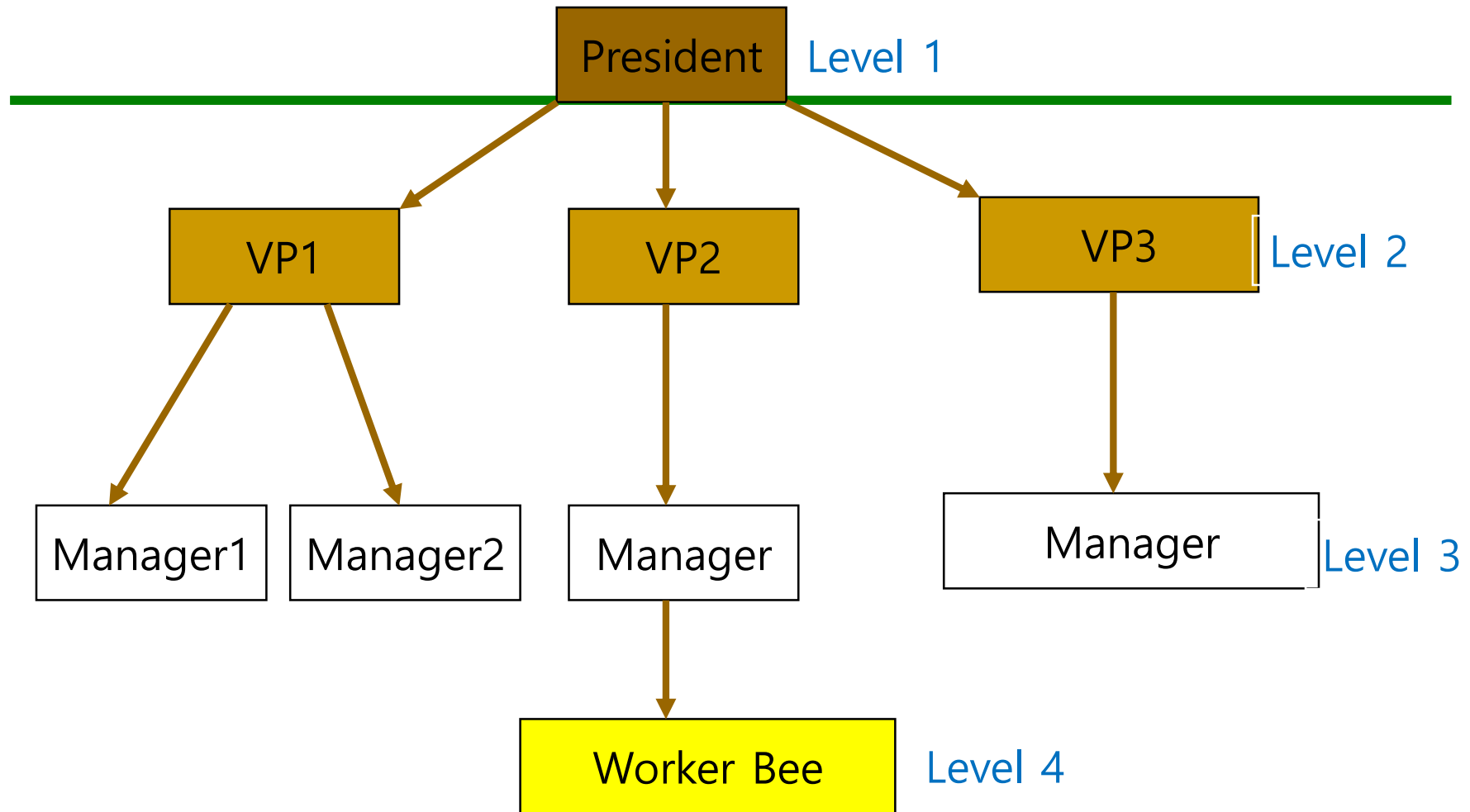
Leaves



Parent, Grandparent, Siblings, Ancestors, Descendants



Levels



- height = depth = number of levels
- Degree of Node = Number Of Children of the Node
- Degree of Tree = Maximum Degree of Node

