

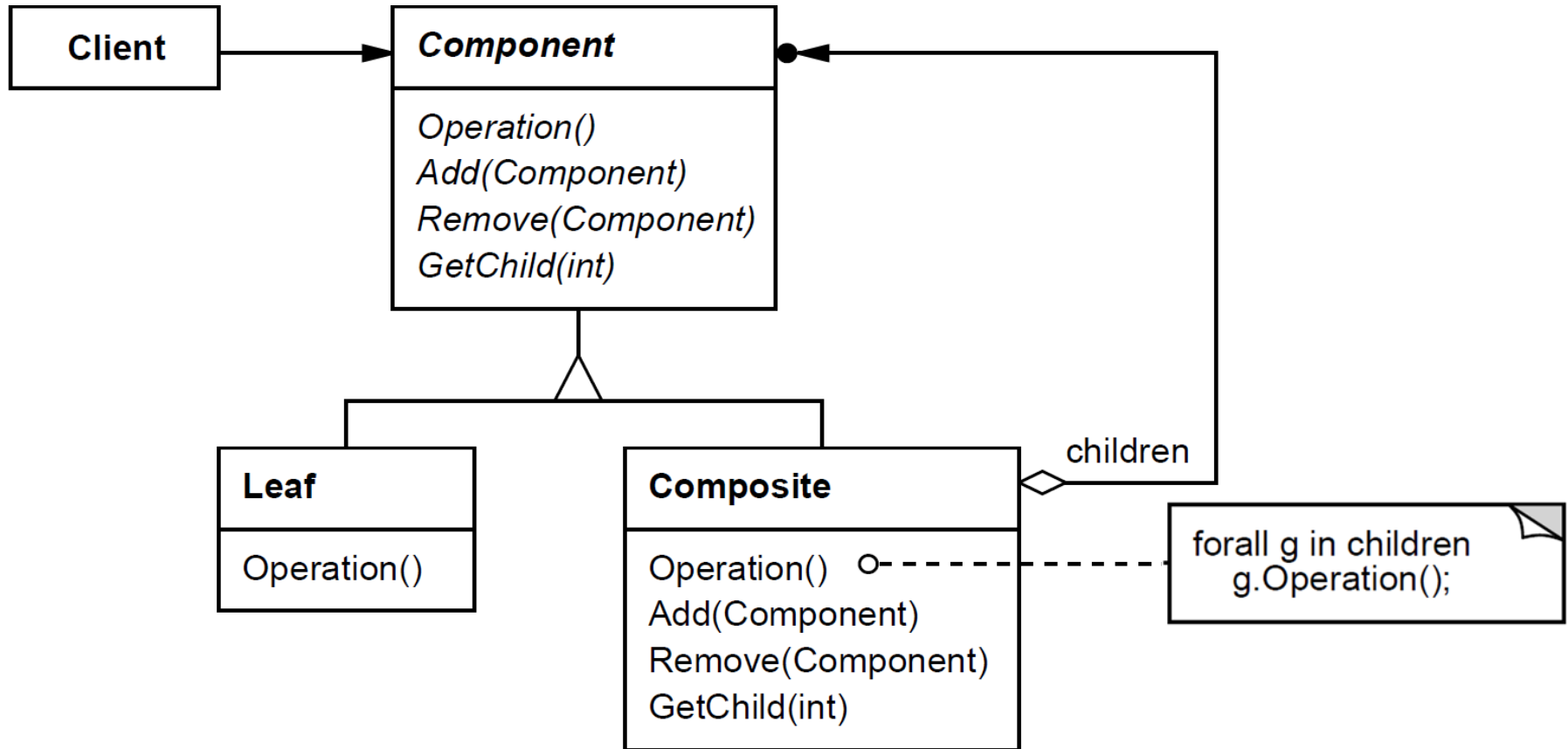
The Composite Pattern

Structure & Functionality

Douglas C. Schmidt

Learning Objectives in This Lesson

- Recognize how the *Composite* pattern can be applied to make the expression tree more uniform & extensible.
- Understand the structure & functionality of the *Composite* pattern.



Douglas C. Schmidt

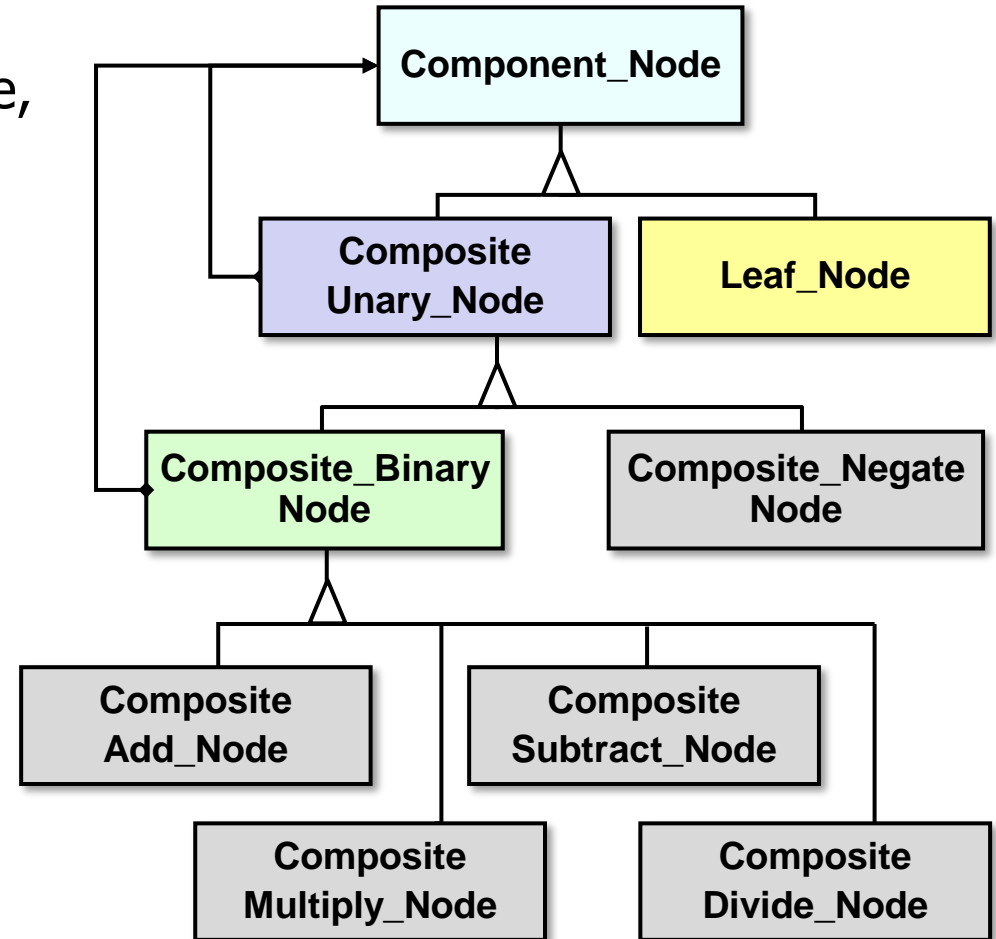
Structure & Functionality of the Composite Pattern

Composite

GoF Object Structural

Intent

- Treat individual objects & multiple, recursively-composed objects uniformly



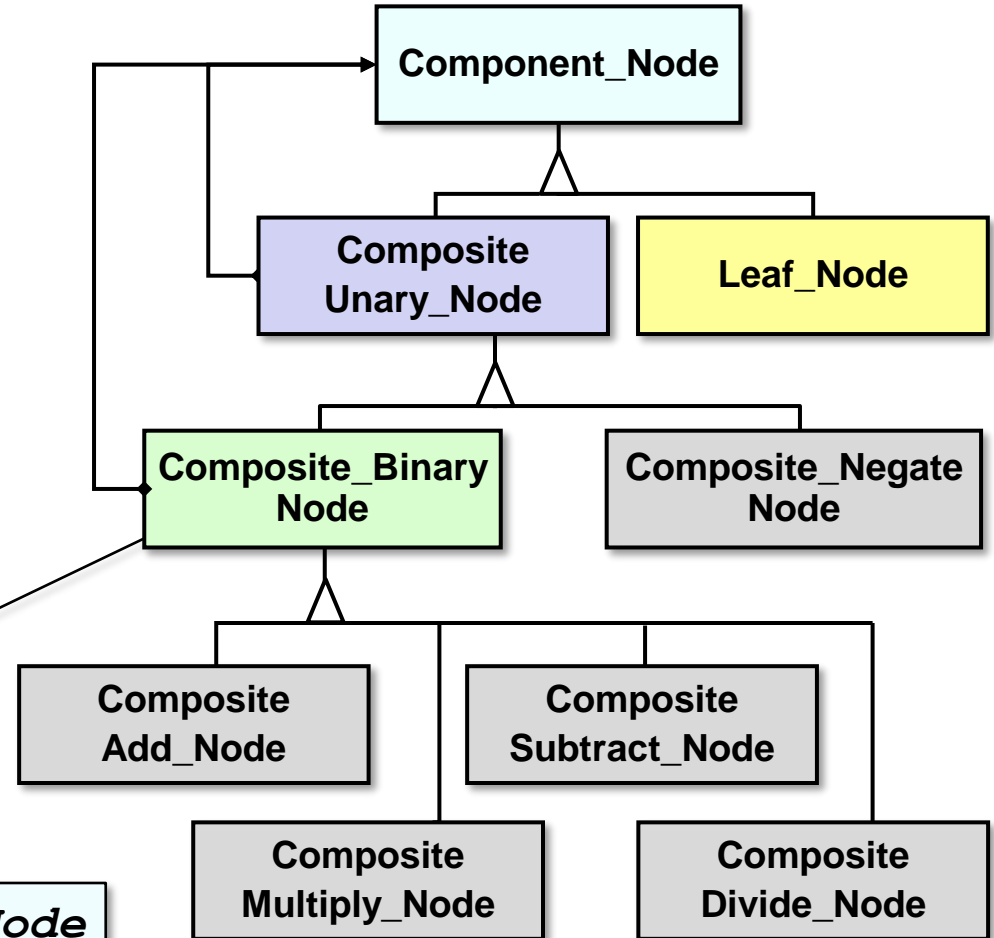
See en.wikipedia.org/wiki/Composite_pattern

Composite

GoF Object Structural

Applicability

- Objects must be composed recursively



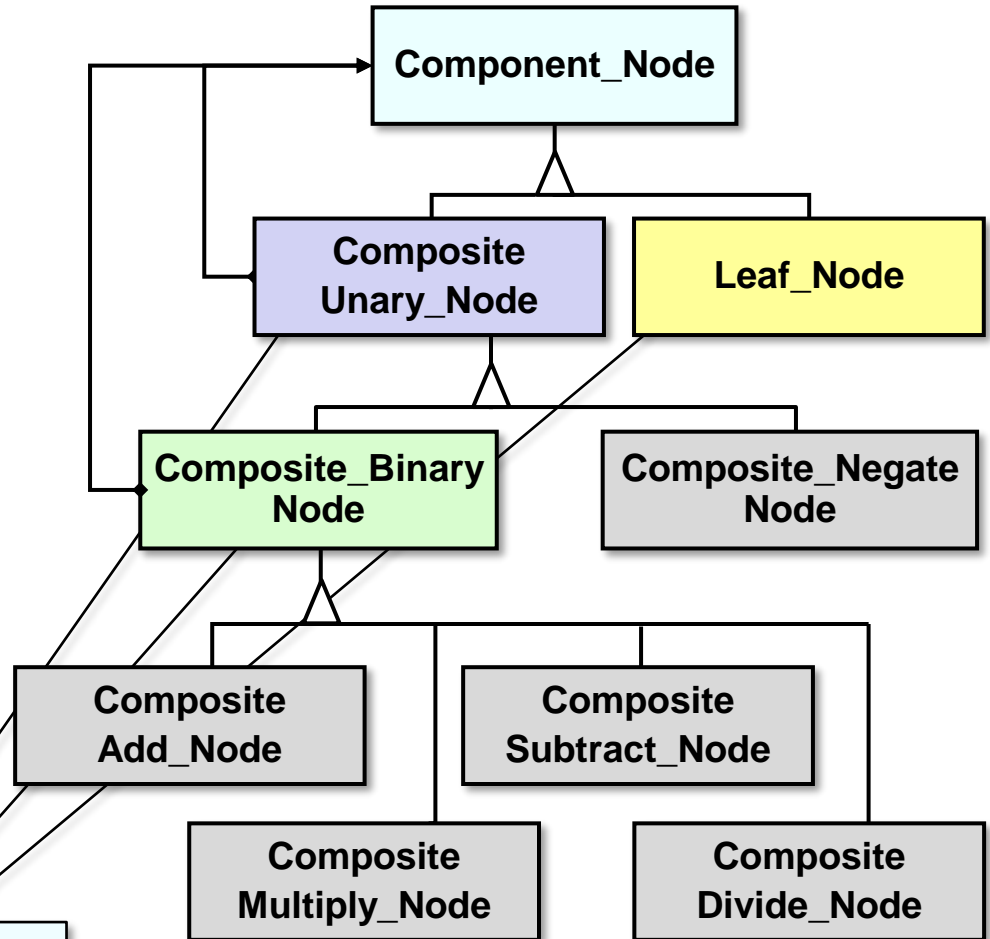
e.g., Composite_Binary_Node contains other types of nodes.

Composite

GoF Object Structural

Applicability

- Objects must be composed recursively
- *And* no distinction between individual & composed elements



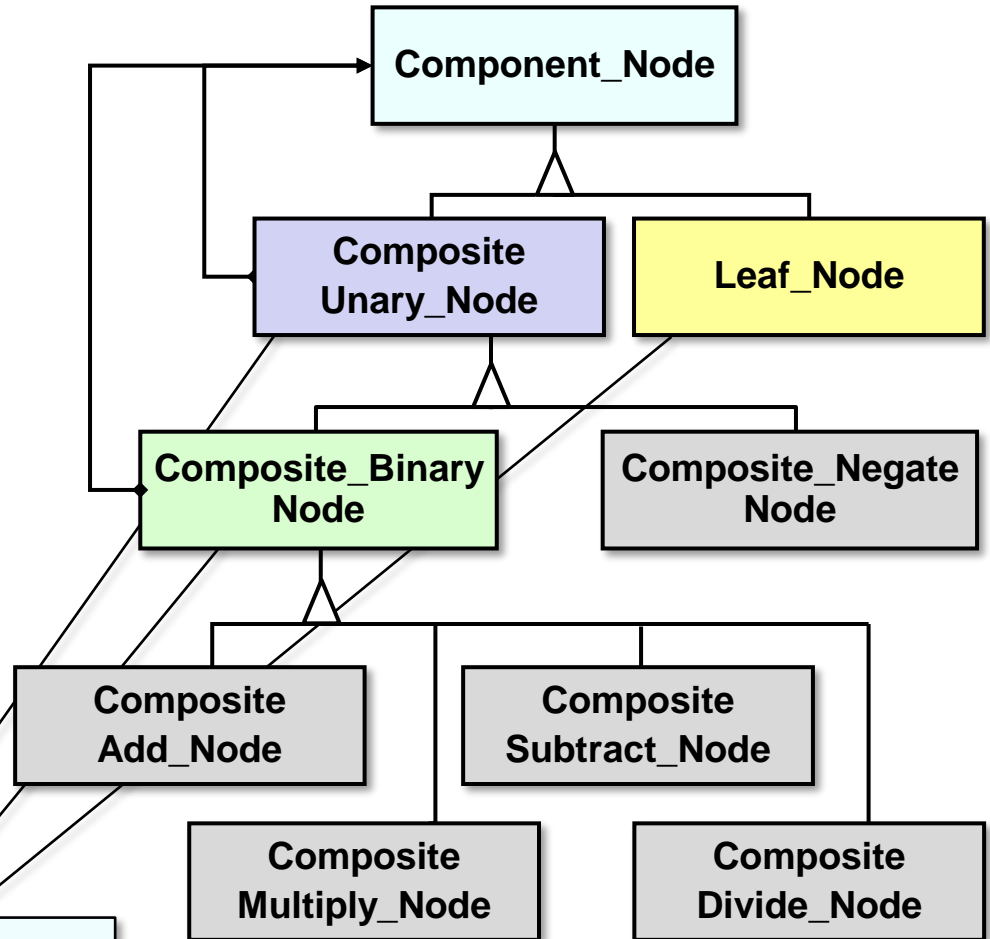
*e.g., Leaf_Nodes & Composite *Nodes all share the same API.*

Composite

GoF Object Structural

Applicability

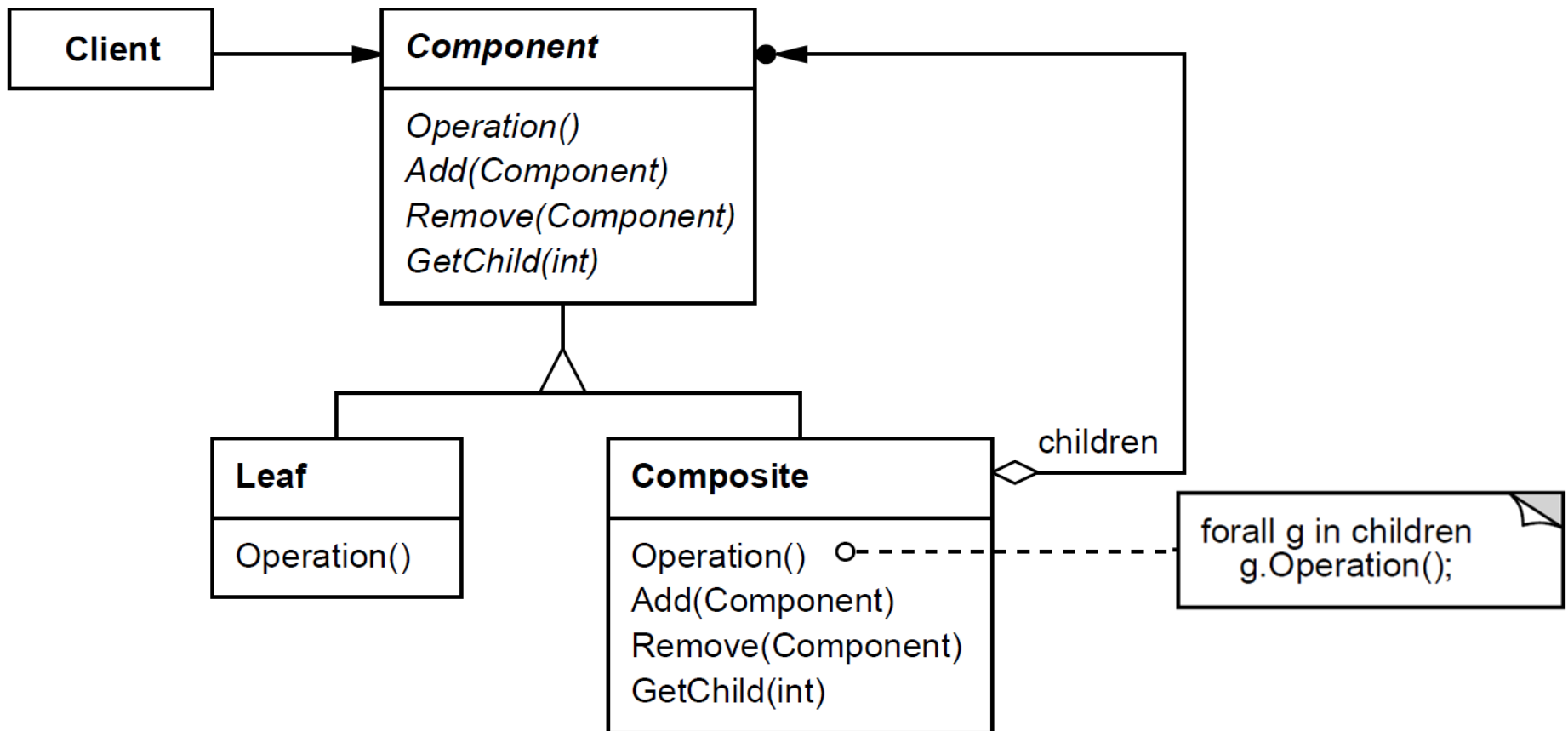
- Objects must be composed recursively
- *And* no distinction between individual & composed elements
- *And* objects in structure can be treated uniformly



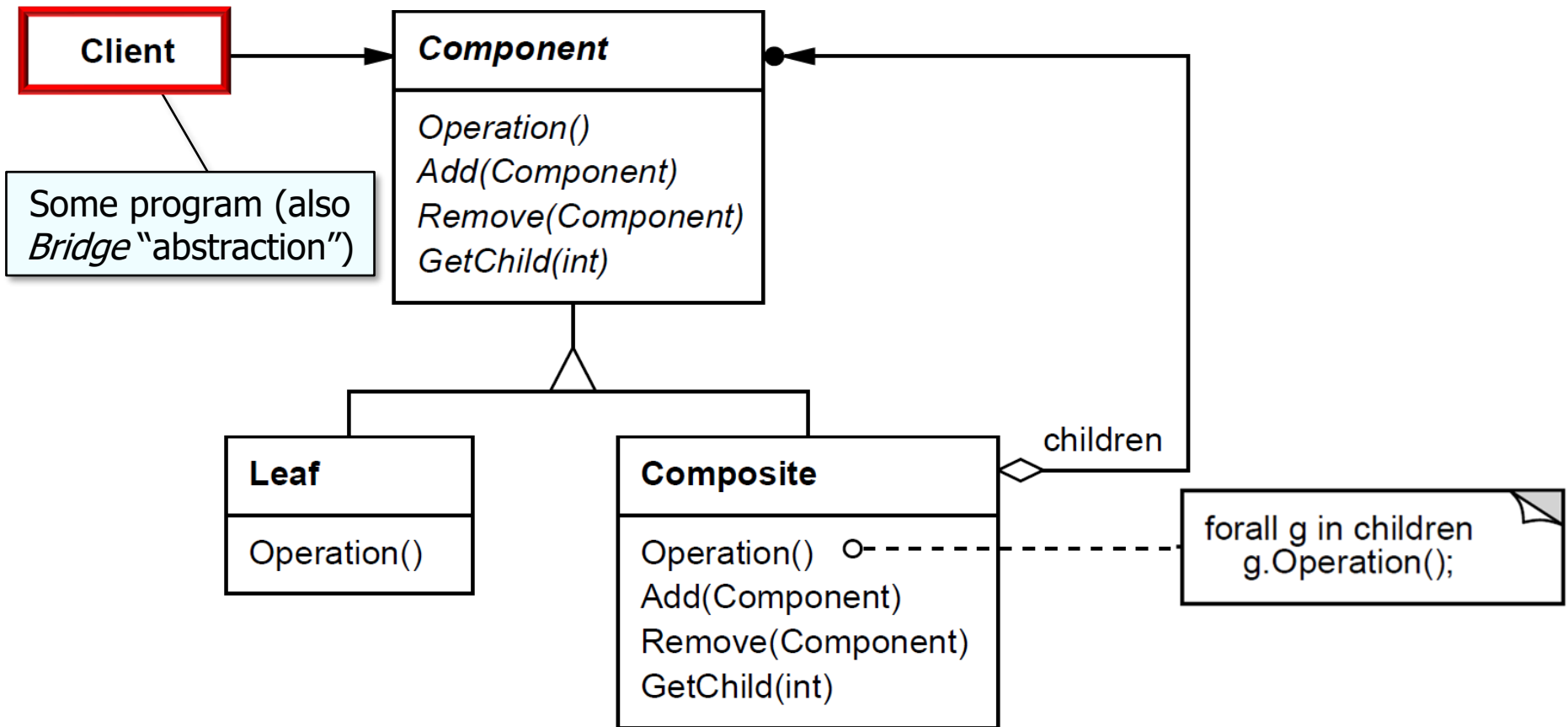
*e.g., Leaf_Nodes & Composite *Nodes are (largely) treated the same by operations on a tree.*

See upcoming lessons on "The *Iterator* Pattern" & "The *Visitor* Pattern."

Structure & participants



Structure & participants

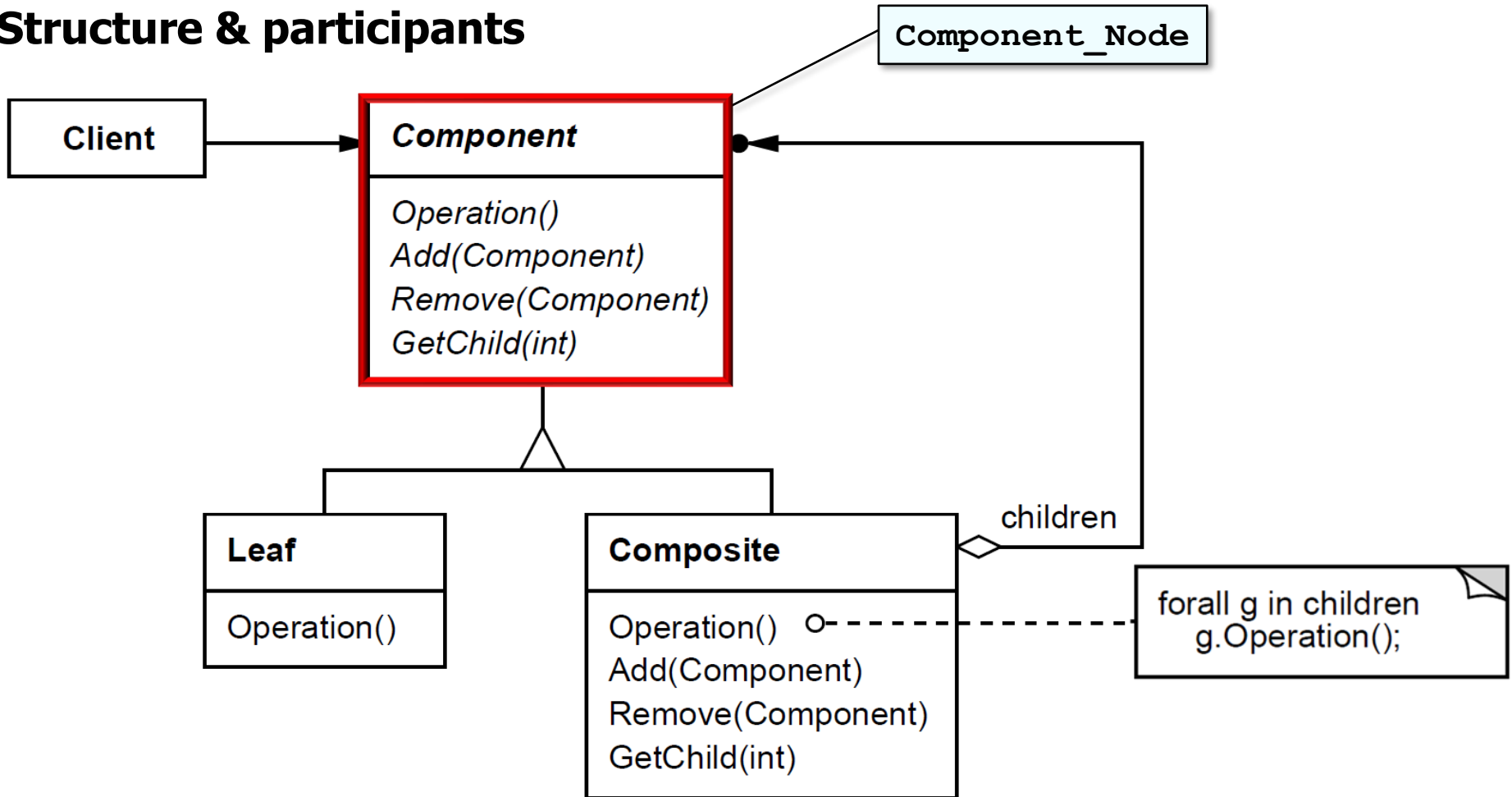


See upcoming lessons on the *Bridge* pattern.

Composite

GoF Object Structural

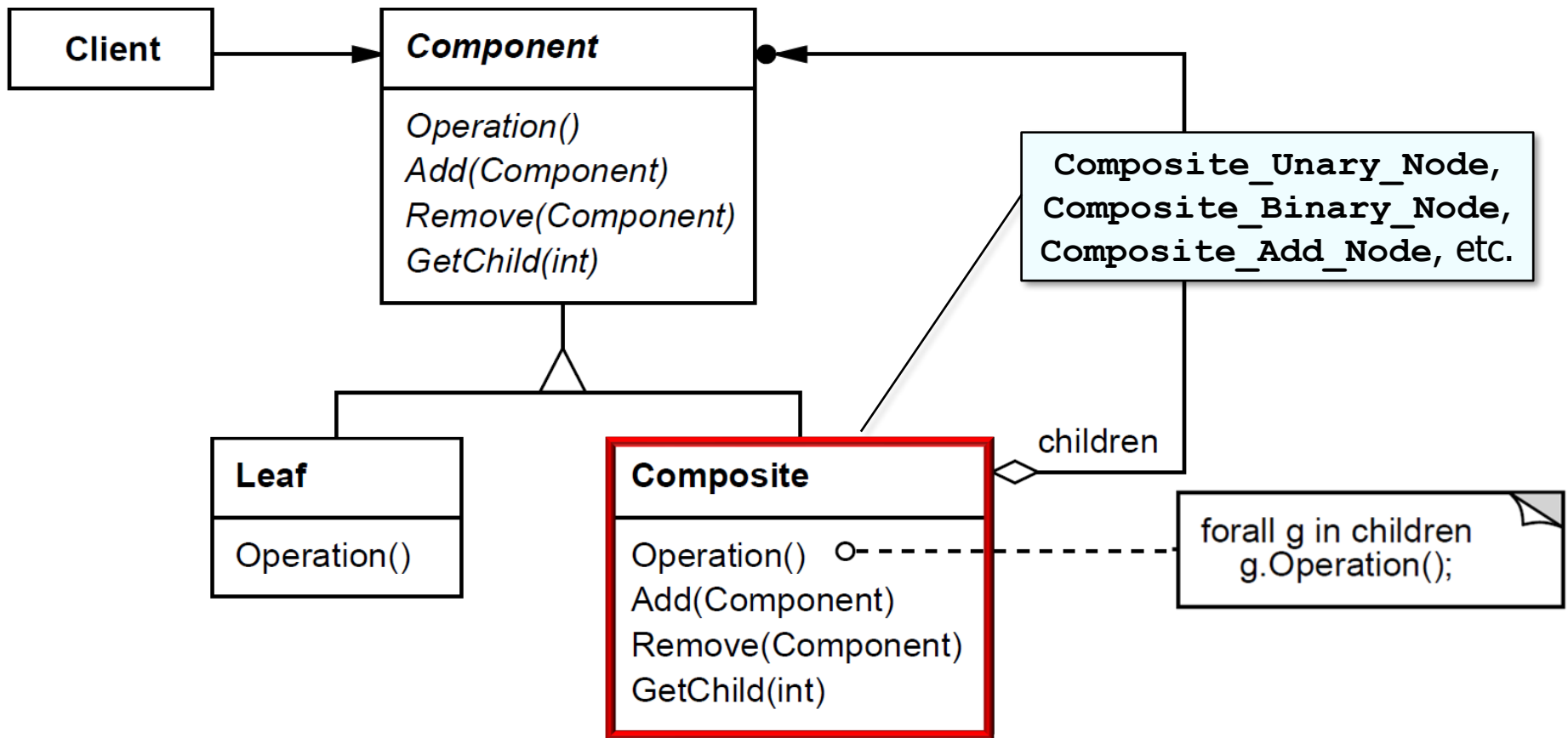
Structure & participants



Composite

GoF Object Structural

Structure & participants



Structure & participants

