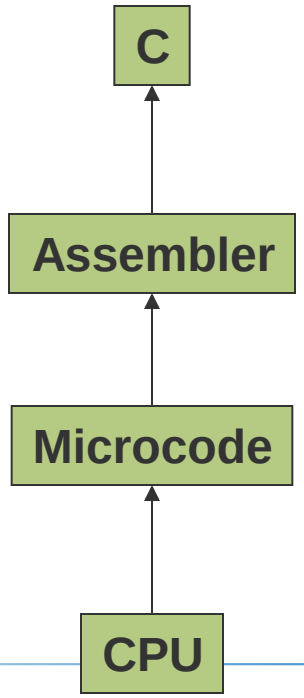
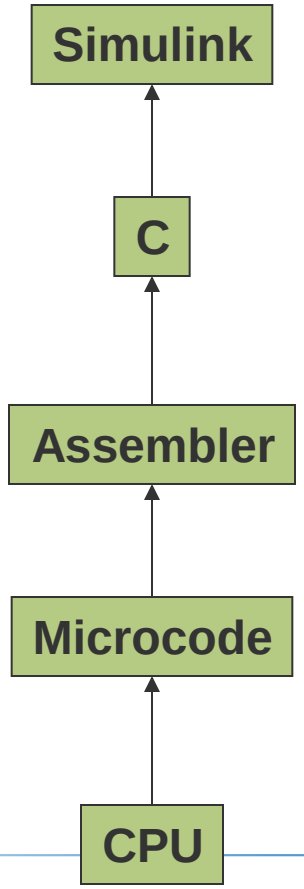
Abstractness, Specificity, and Complexity in Software Design

Stefan Wagner and Florian Deißenböck
Technische Universität München, Germany

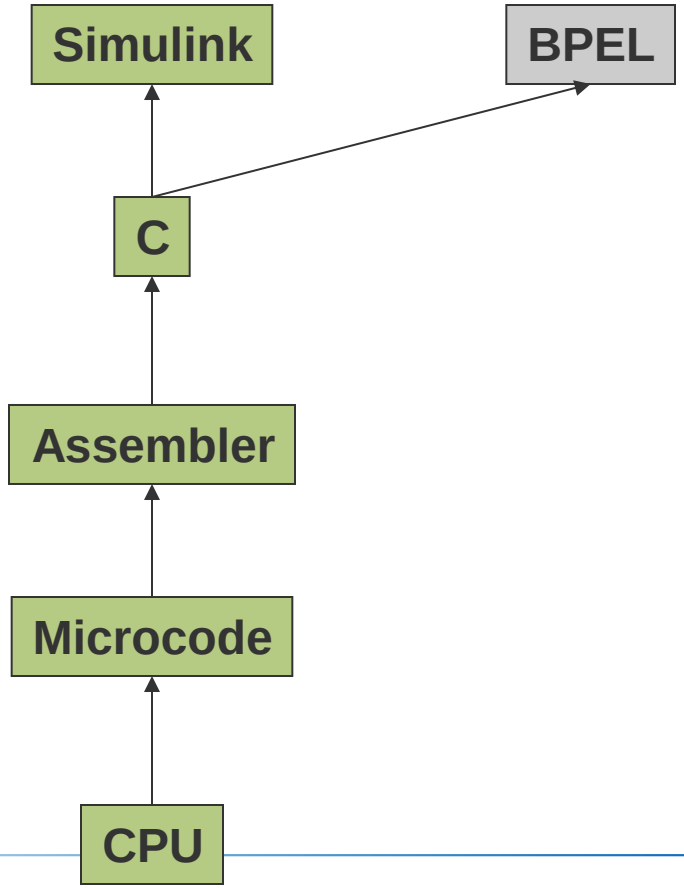
Abstraction Examples



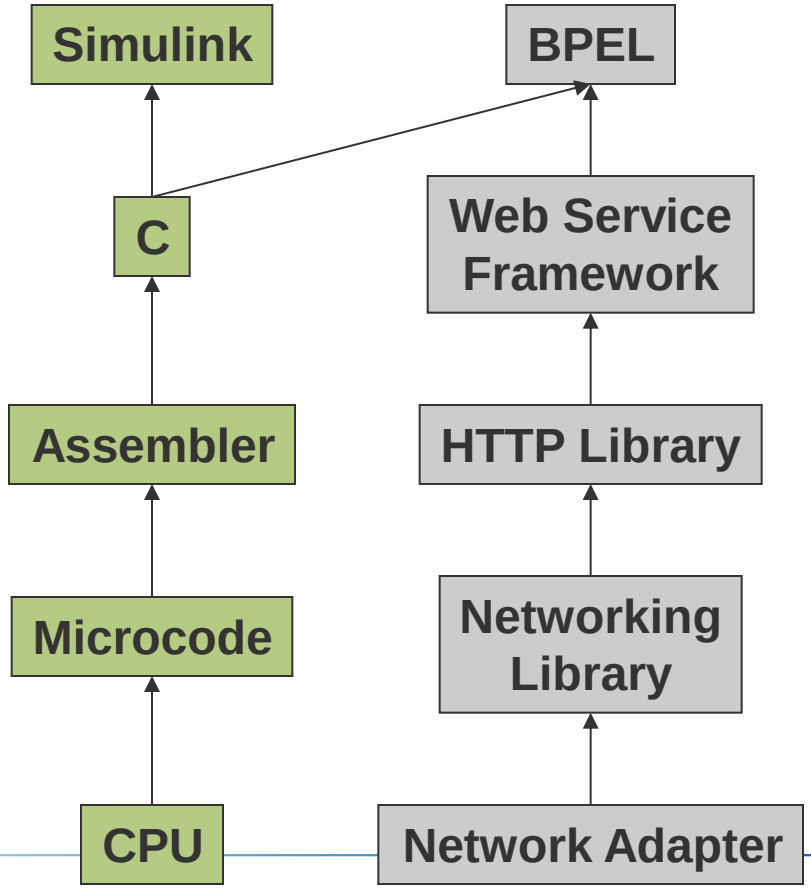
Abstraction Examples



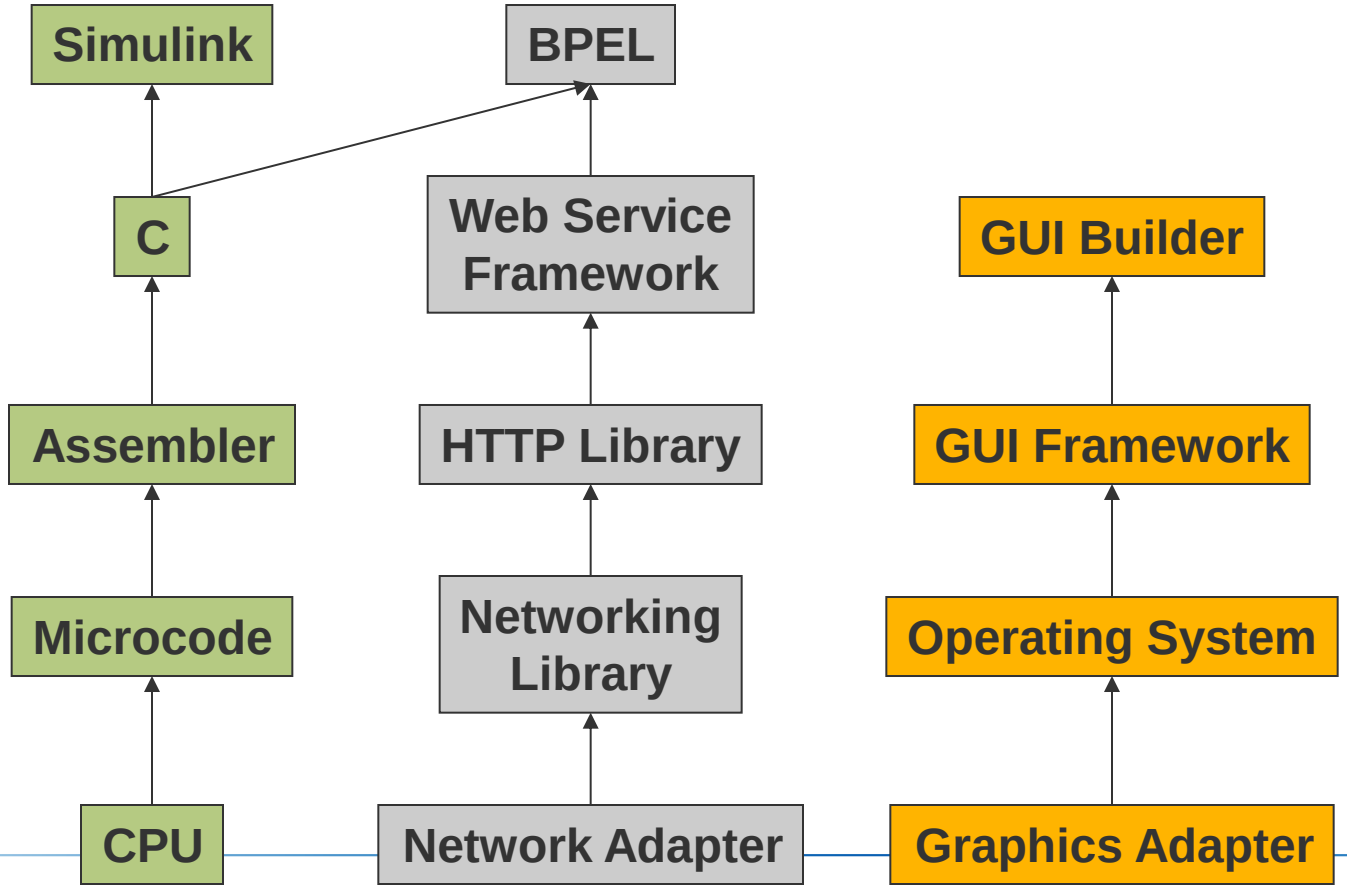
Abstraction Examples



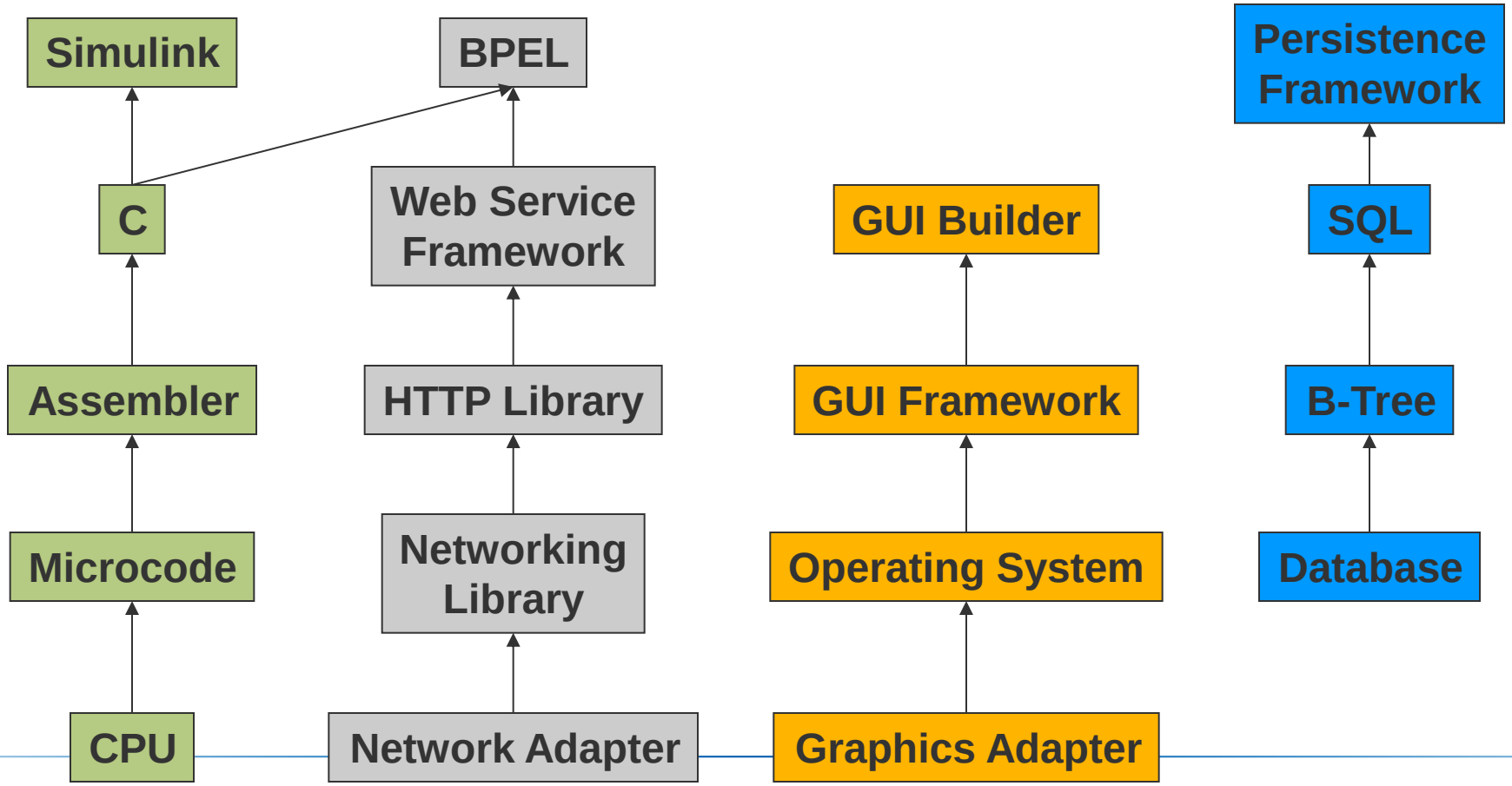
Abstraction Examples



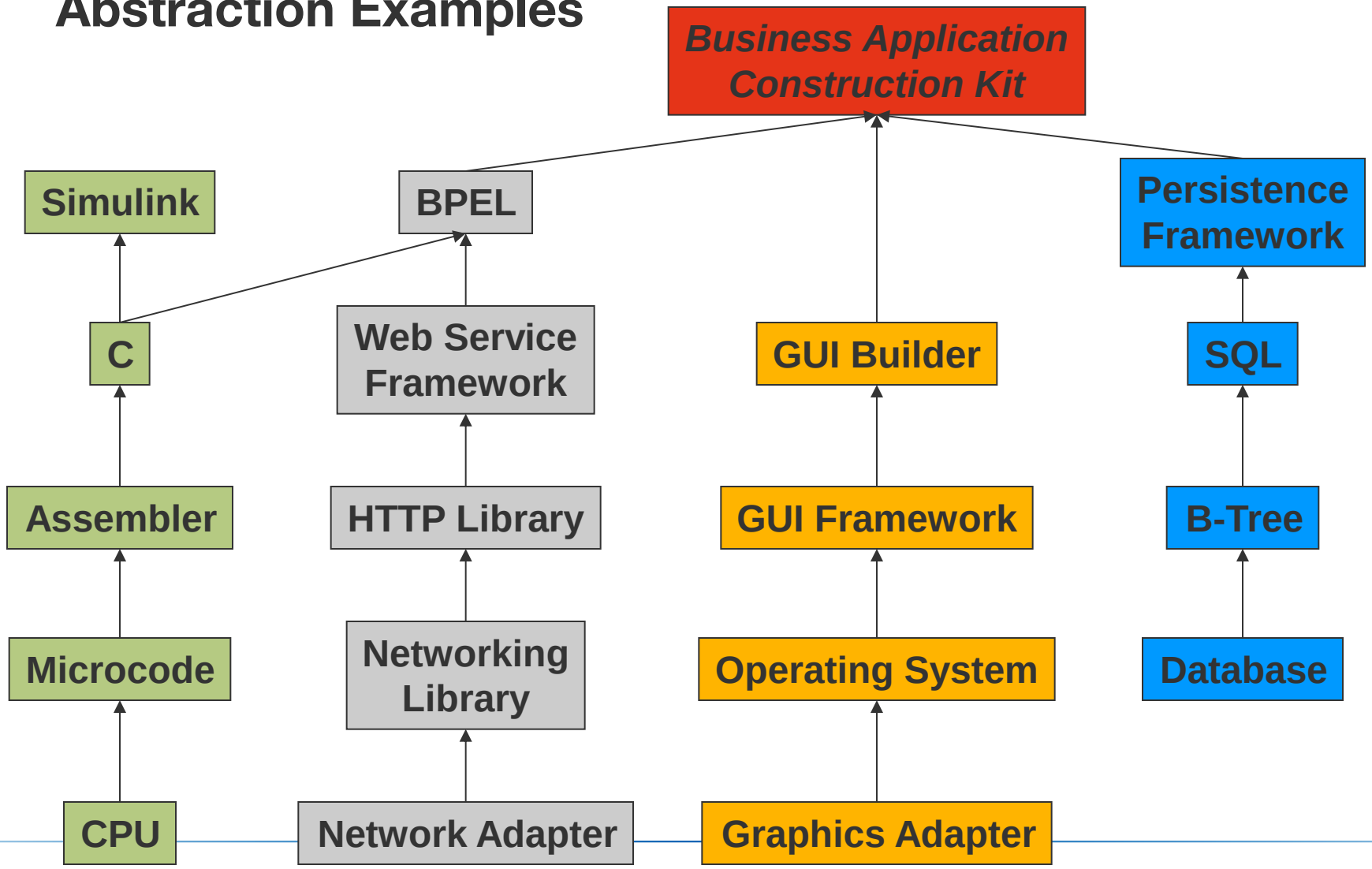
Abstraction Examples



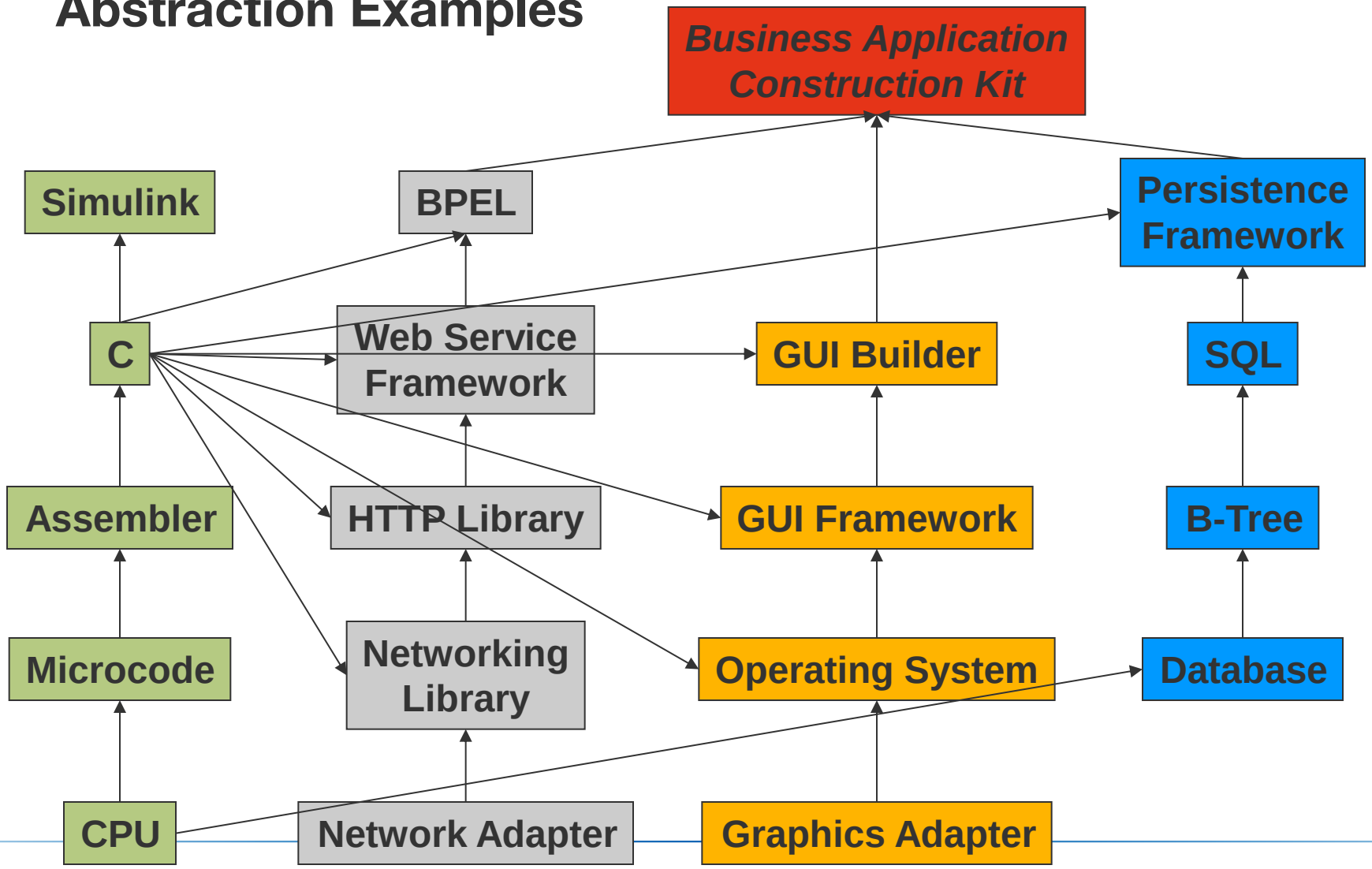
Abstraction Examples



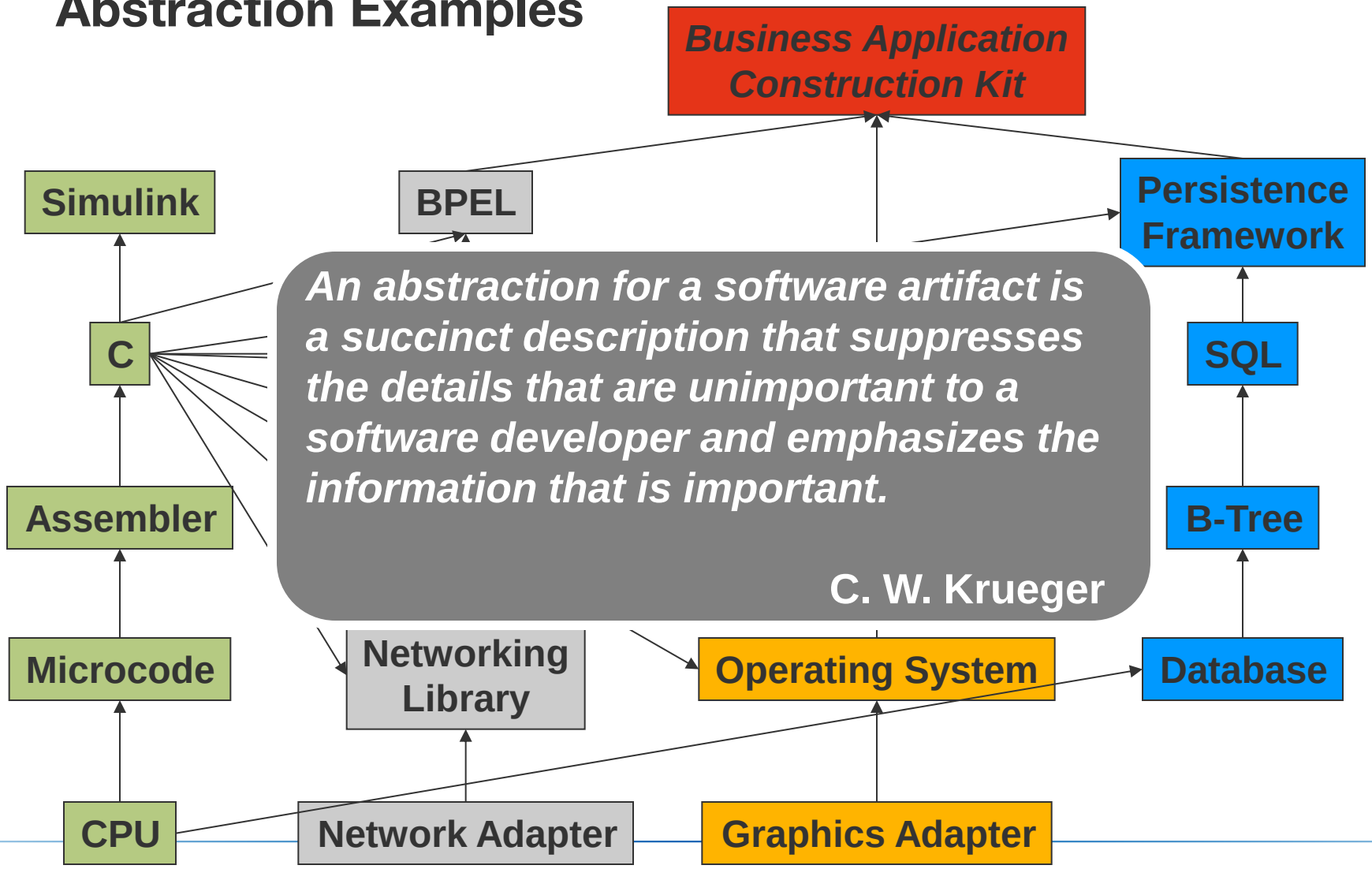
Abstraction Examples



Abstraction Examples



Abstraction Examples

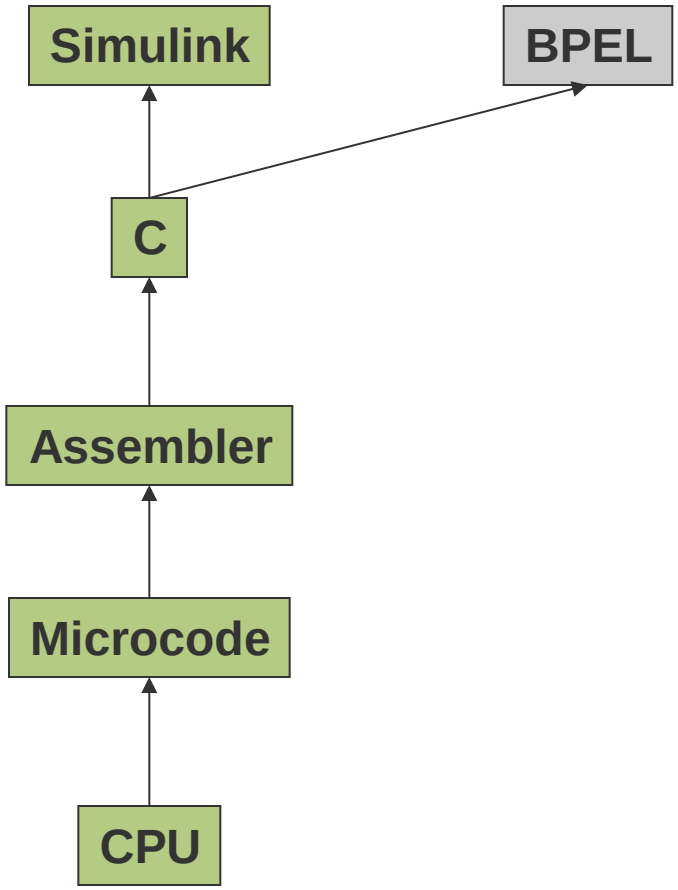


An abstraction for a software artifact is a succinct description that suppresses the details that are unimportant to a software developer and emphasizes the information that is important.

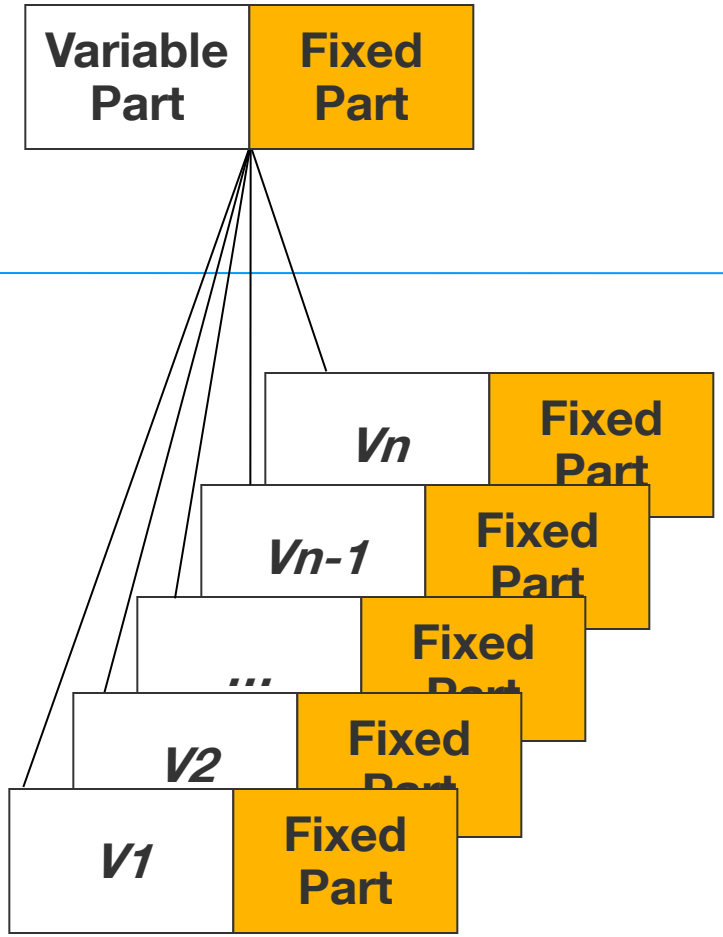
C. W. Krueger

Problem Statement

- Abstraction is of paramount importance for software development.
- What are the implications on specificity?
- What are the implications on complexity?
- What are the implications and trade-offs in abstractions?
- How to design and use abstractions?



Two Levels of Abstraction

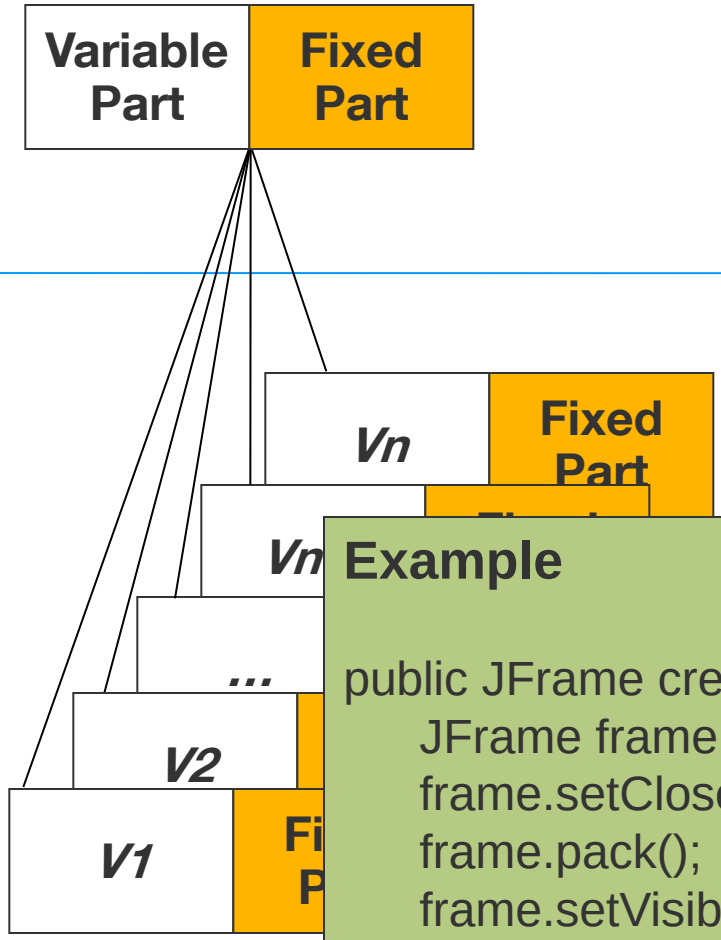


Abstraction Specification

Abstraction Realization

[C. W. Krueger. Software Reuse. 1992]

Two Levels of Abstraction



Example

```
public JFrame createWindow(String title) {
    JFrame frame = new JFrame(title);
    frame.setCloseOp(EXIT_ON_CLOSE);
    frame.pack();
    frame.setVisible(true);
    return frame;
}
```

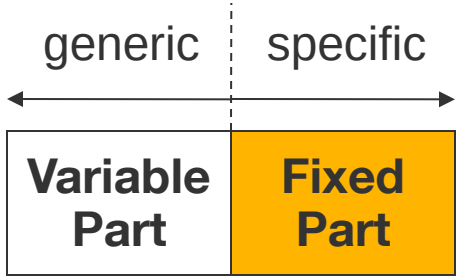
Abstractness & Specificity

Abstractness

- Abstraction means information loss
- Remove explicit detail ⇨ model building
- Defined by amount of variable information

Specificity

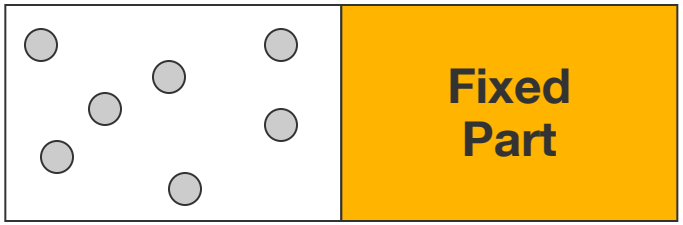
- Defined by the number of contexts it can be used in
- The larger the variable part, the more generic
- The larger the fixed part, the more specific



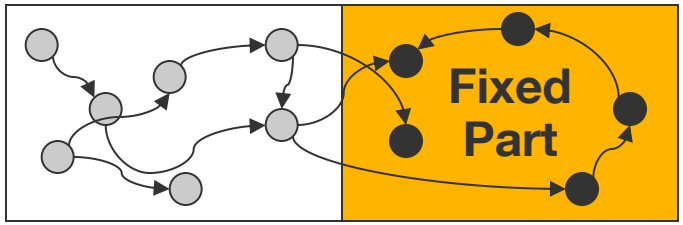
Complexity

- Complexity reduction is a main goal of abstraction
- Various definitions (even philosophy has not a unique one)
- Detail complexity*: Number of parts
- Dynamic complexity*: Cause and effect relationships

Detail Complexity

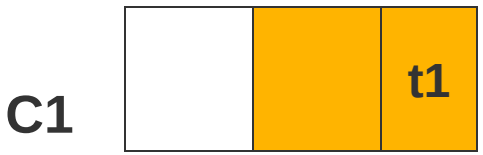


Dynamic Complexity



* [Peter M. Senge. The Fifth Discipline. 1990]

Types of Abstraction

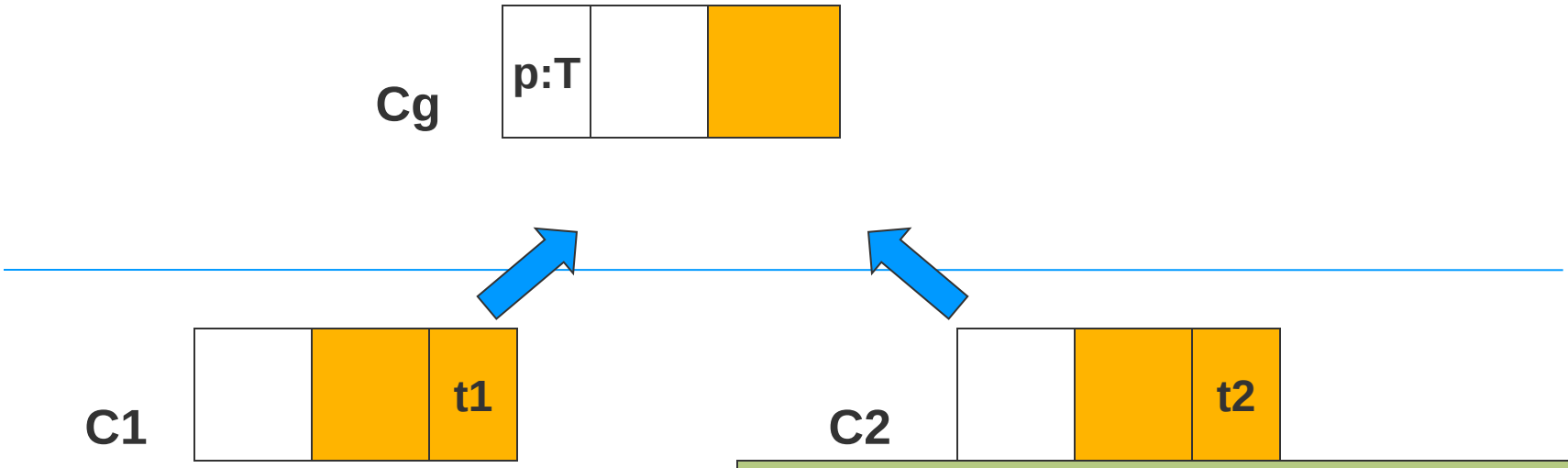


Types of Abstraction


```
JFrame f = new JFrame("Choose Source Fil
f.setCloseOp(EXIT_ON_CLOSE);
…
```

```
JFrame f = new JFrame("Choose Target File");
f.setCloseOp(EXIT_ON_CLOSE);
…
```

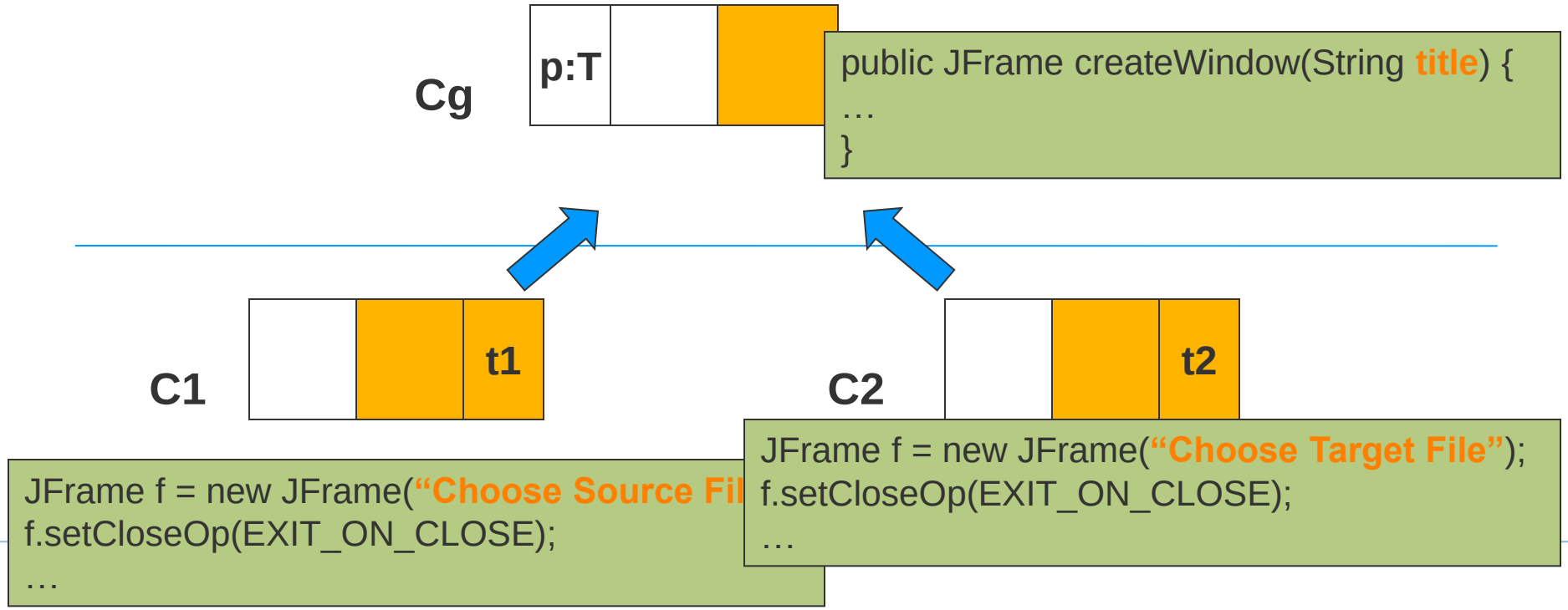
Types of Abstraction



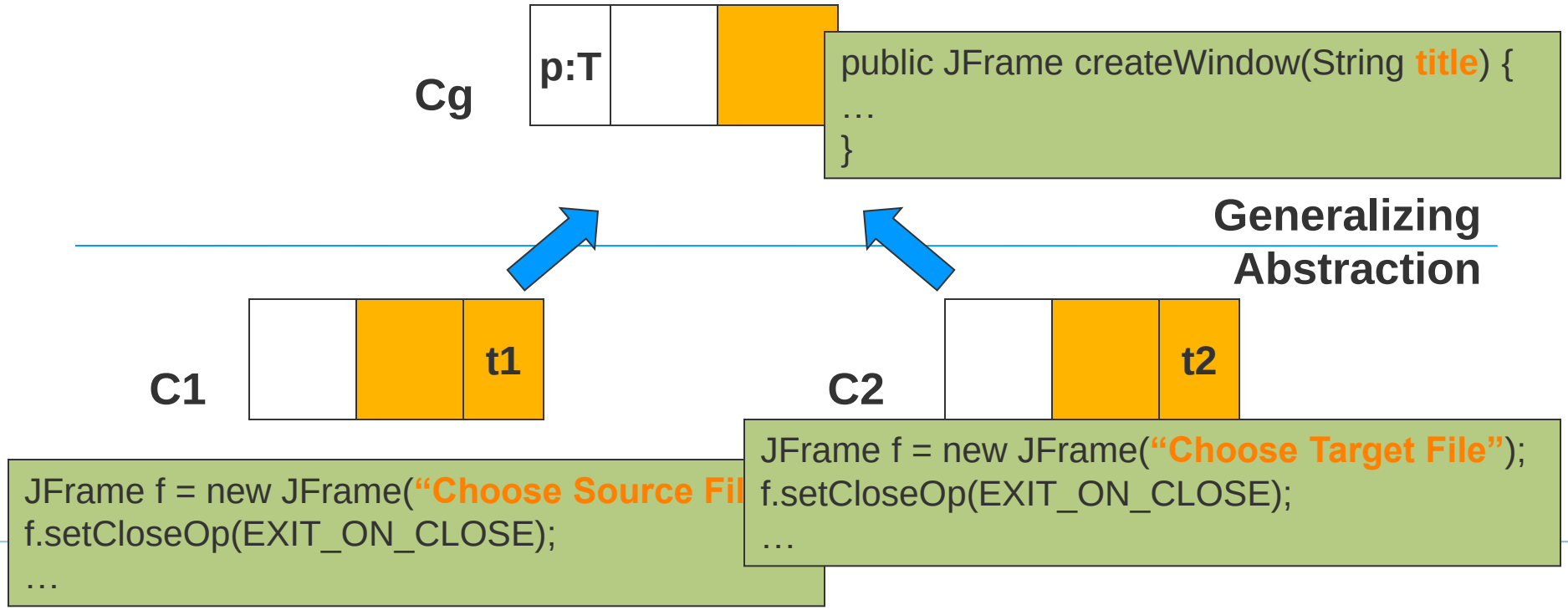

```
JFrame f = new JFrame("Choose Source Fil
f.setCloseOp(EXIT_ON_CLOSE);
…
```

```
JFrame f = new JFrame("Choose Target File");
f.setCloseOp(EXIT_ON_CLOSE);
…
```

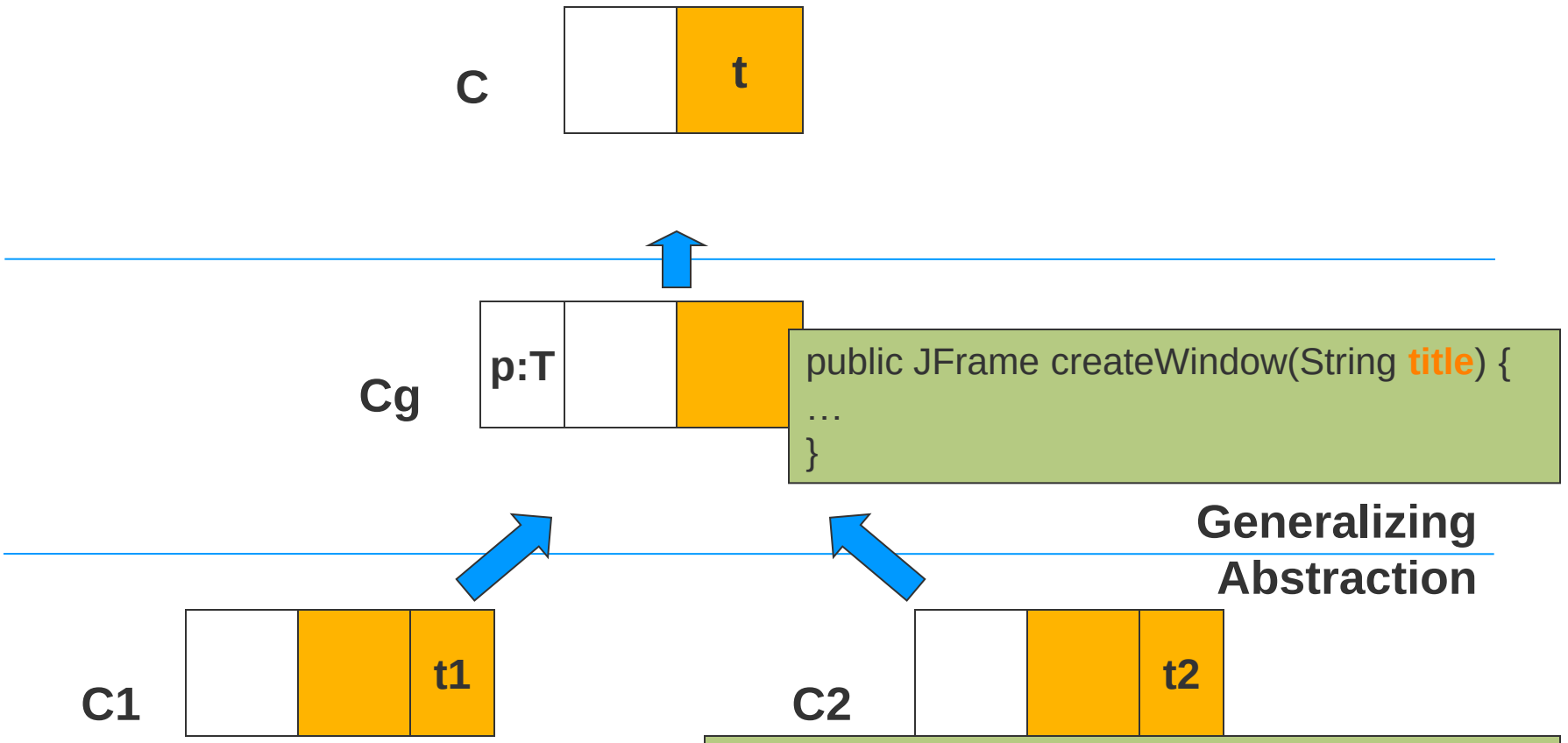
Types of Abstraction



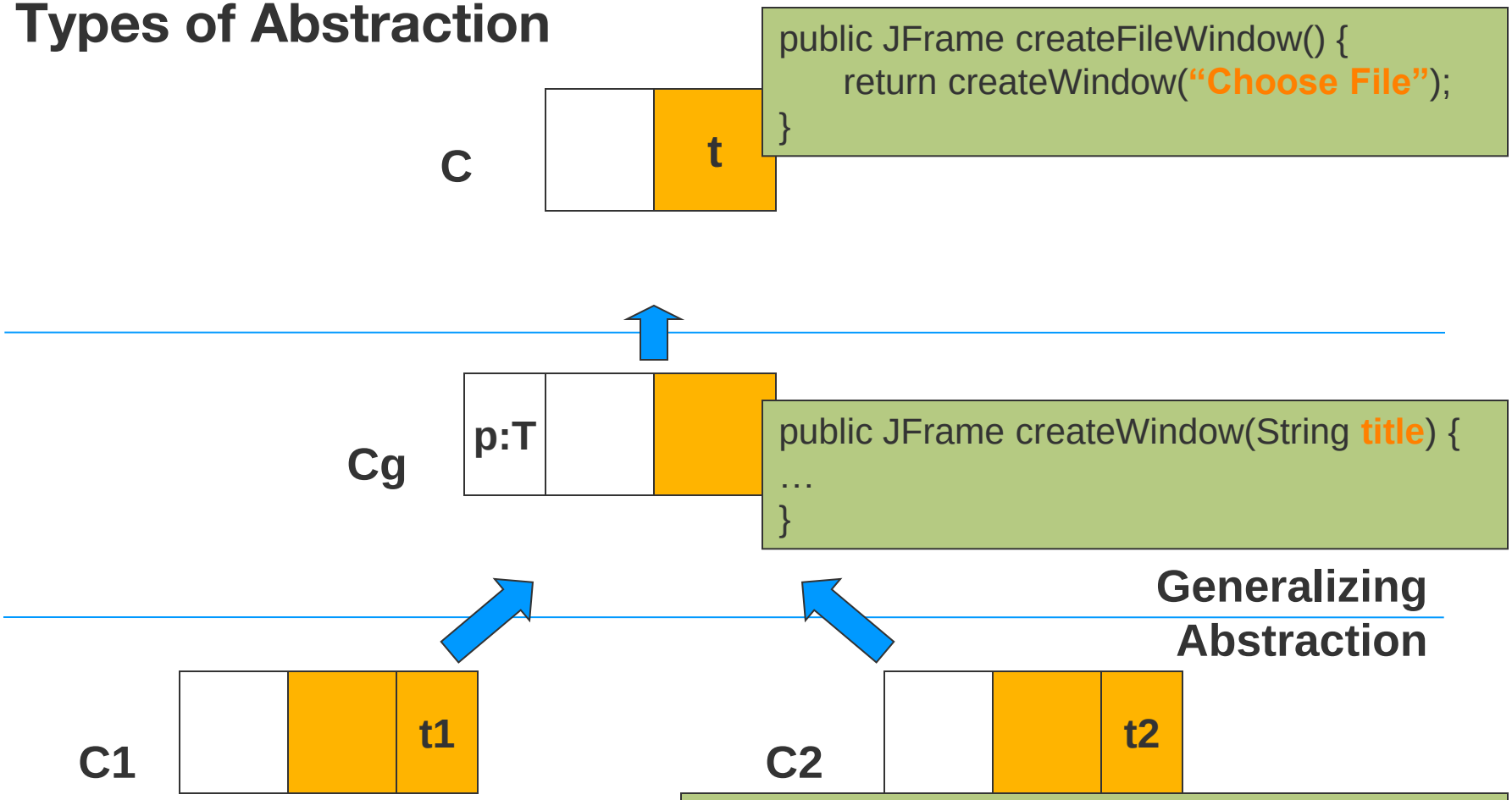
Types of Abstraction



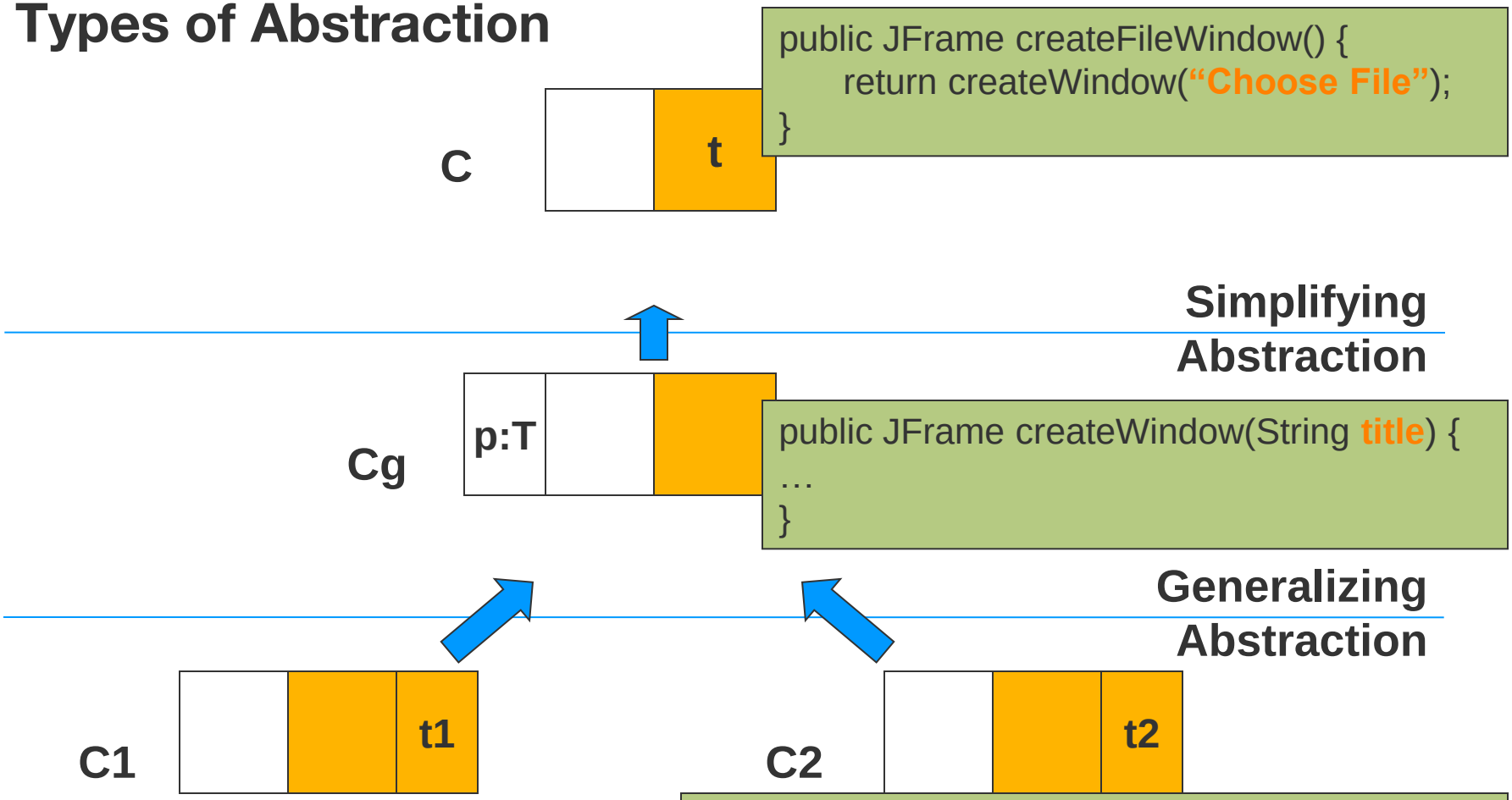
Types of Abstraction



Types of Abstraction

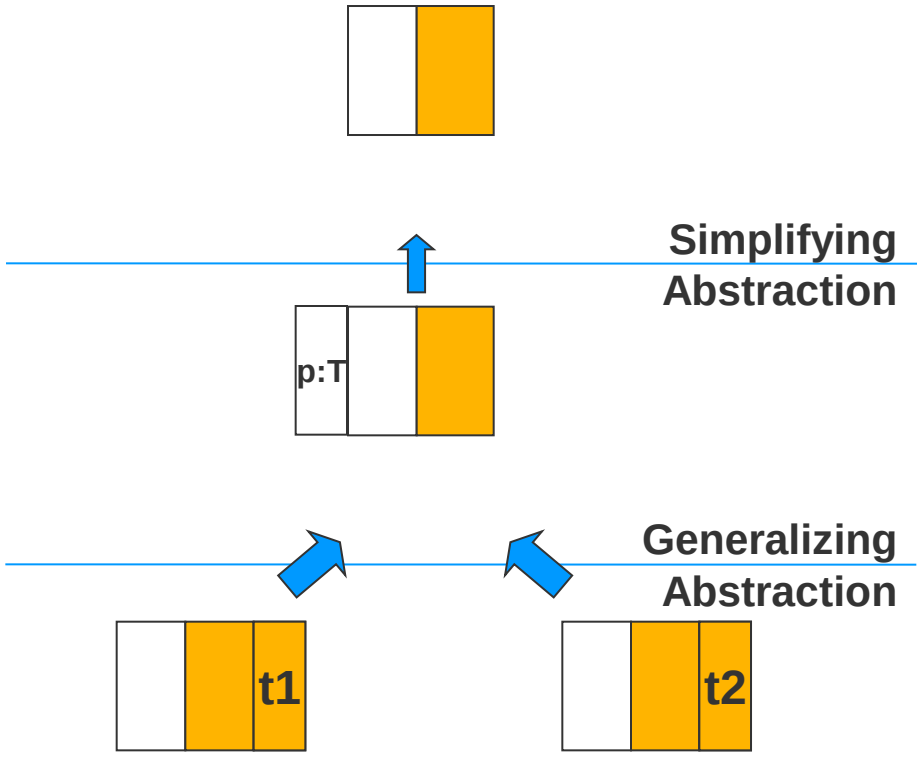


Types of Abstraction



The Influences of Abstraction

Type	Specificity	Detail Complexity	Dynamic Complexity
Genera-lizing	-	0/-	+
Simpli-fying	+	-	+/-



Consequences

1. Generalizing abstraction increases dynamic complexity
2. Simplifying abstraction increases specificity
3. To manage complexity, design specific and generic

Consequences

1. **Generalizing abstraction increases dynamic complexity**
2. **Simplifying abstraction increases specificity**
3. **To manage complexity, design specific and generic**

Example
Swing JFrame can be used by setting only one of its > 65 parameters.

Conclusions

- Motivation
  - Abstraction is an essential activity in software engineering
  - Used to increase comprehensibility and reuse
  - Effects are rarely discussed and not well understood
- Contribution
  - Two basic types of abstraction: simplifying and generalizing
  - Influences on specificity and complexity
- Future work
  - Developing more examples
  - Formal framework
  - Relation to redundancy?