The Opus Research Group

Identifying Design Problems with Code Smell Agglomerations

Basic Concepts

Benedicte Agbachi, Eduardo Fernandes, and Alessandro Garcia

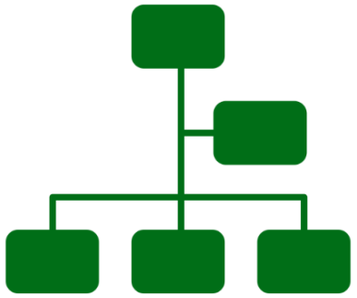
OPUS Research Group

LES | DI | PUC-Rio - Brazil

Basics of
Software Design

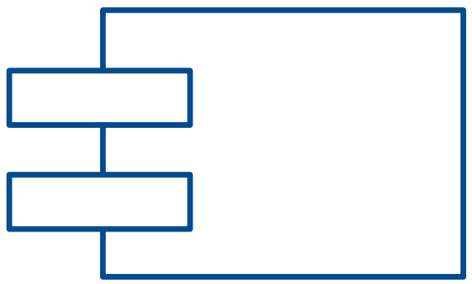
{oagbachi, emfernandes, afgarcia}@inf.puc-rio.br

Software design is…



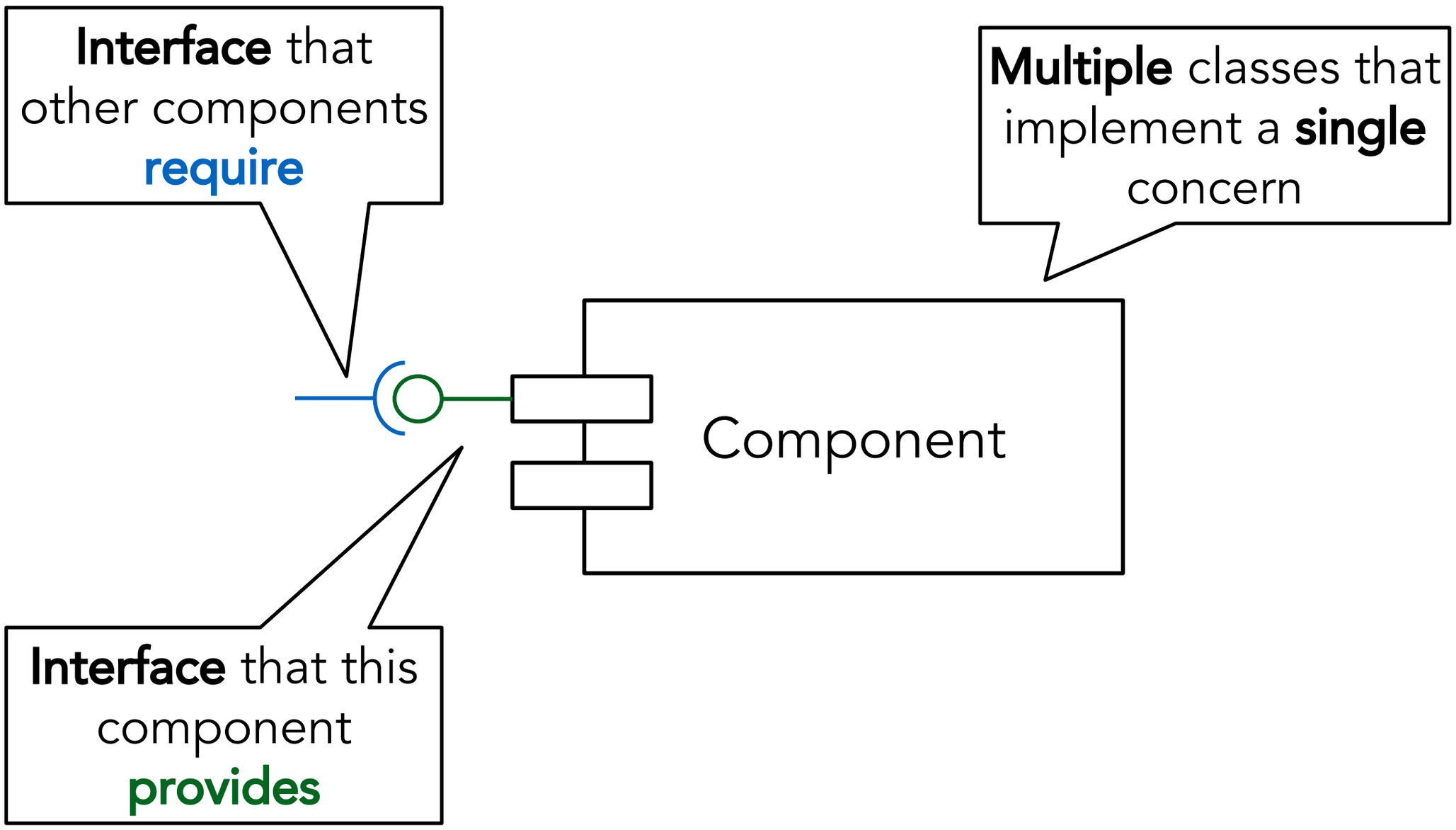
Organization of concerns (features) in a program

Components and their **relationships**

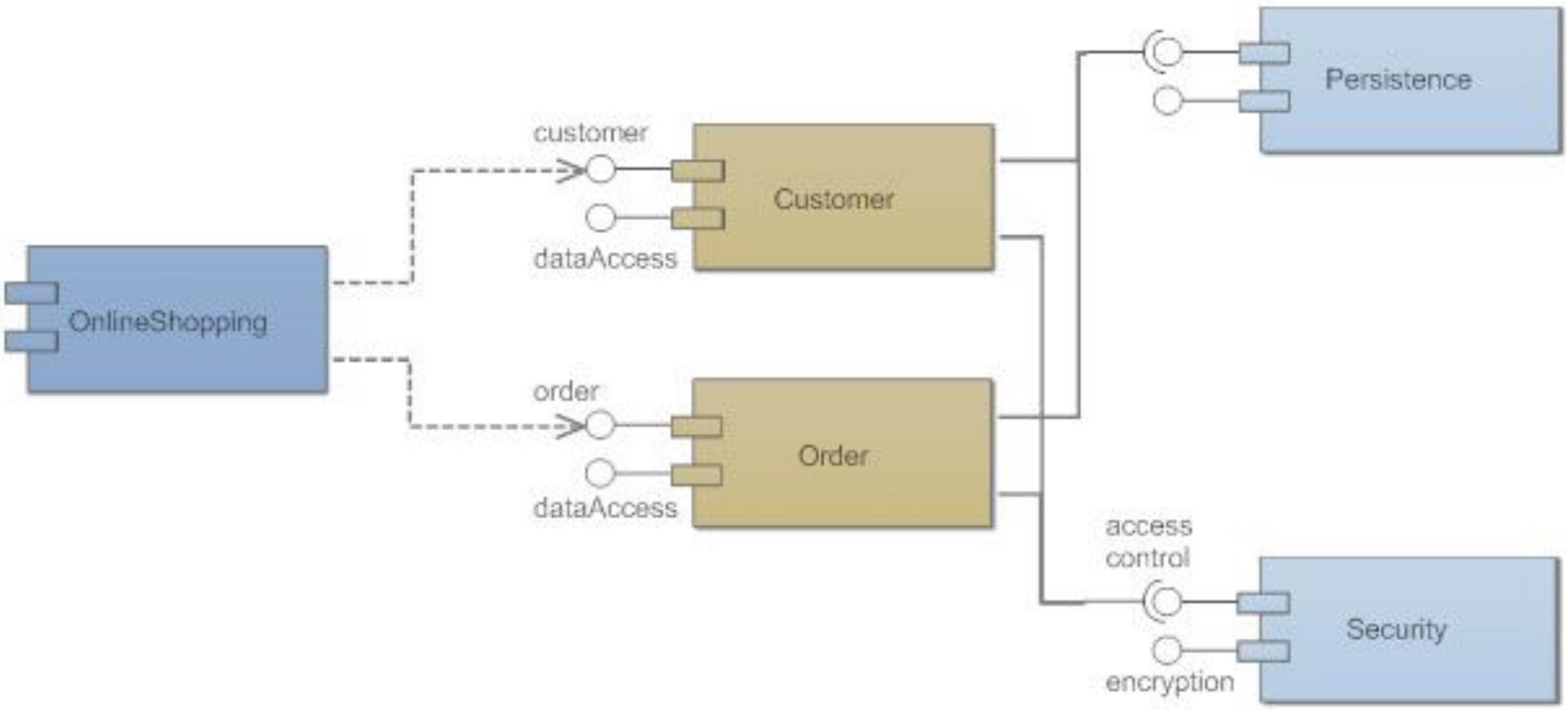


{oagbachi, emfernandes, afgarcia}@inf.puc-rio.br

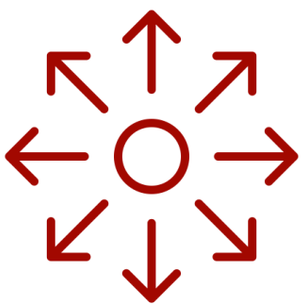
Notation of software design



An example of software design



Cross-cutting concern



One feature scattered in **several** components

It **cross-cuts** other concerns, just like an “intruder”

{oagbachi, emfernandes, afgarcia}@inf.puc-rio.br

Design Principles

versus

Design Problems

Design principles

How to **well design** components and relationships

Aimed at **reducing** efforts with maintenance

{oagbachi, emfernandes, afgarcia}@inf.puc-rio.br

Some well-known design principles

Open-Closed	A class should be **extensible** without need to change it
Single responsibility	Each class should have only **one reason to change**
Interface segregation	Each interface should target a **specific type of client** components

Design problems are caused by…

Violations of well-known design principles

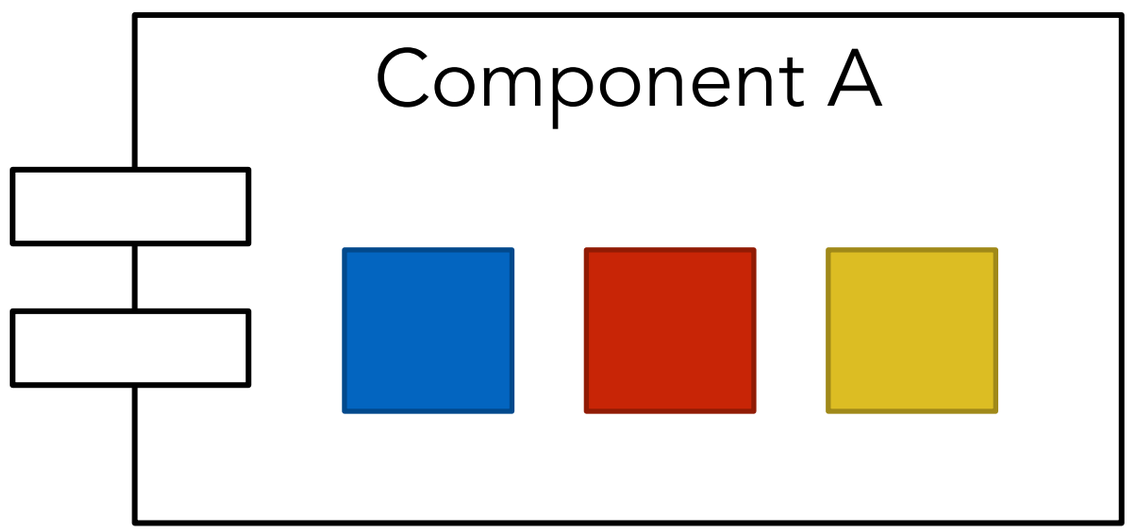
Unintended decisions that violate the original software design

{oagbachi, emfernandes, afgarcia}@inf.puc-rio.br

Example 1: Component Concern Overload

Ideal case: Component A should implement only a single concern

Component concern overload: A implements several concerns



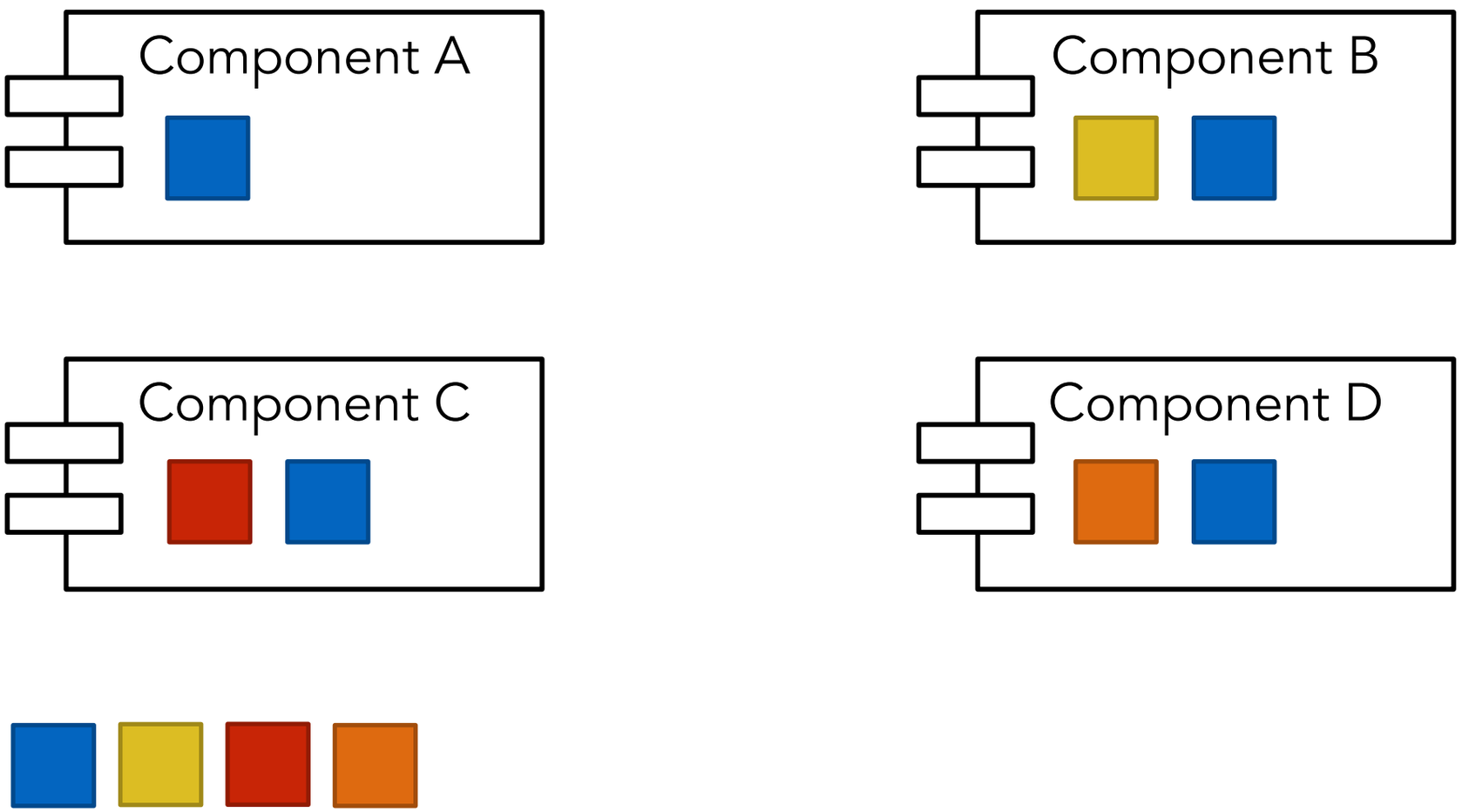
Concerns:

{oagbachi, emfernandes, afgarcia}@inf.puc-rio.br

Example 2: Scattered Concern

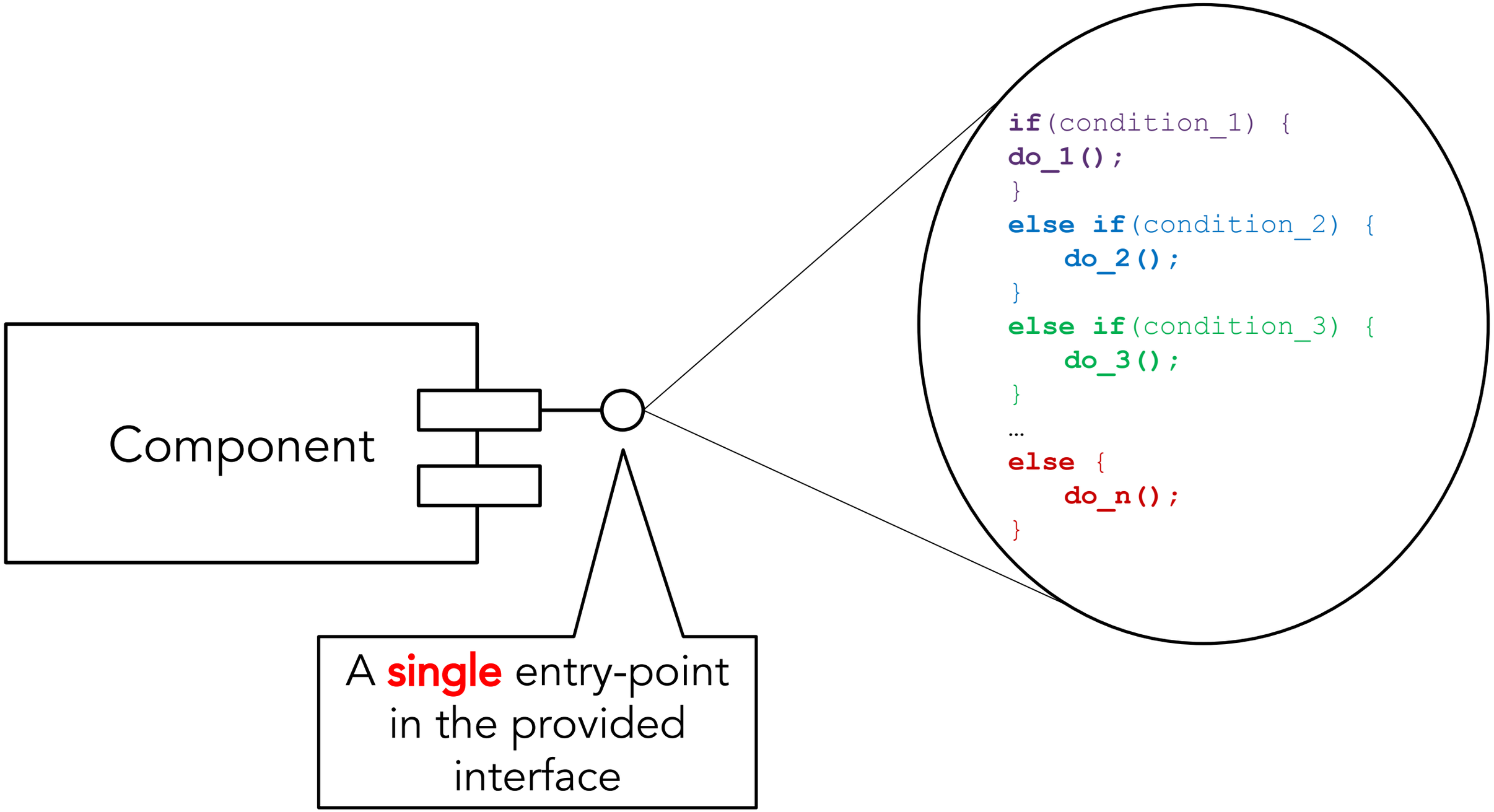
Scattered concern: not only Component A implements , but also B, C, and D

Ideal case: **one** concern per component



{oagbachi, emfernandes, afgarcia}@inf.puc-rio.br

Example 3: Ambiguous Interface



{oagbachi, emfernandes, afgarcia}@inf.puc-rio.br

Code Smells and Agglomerations

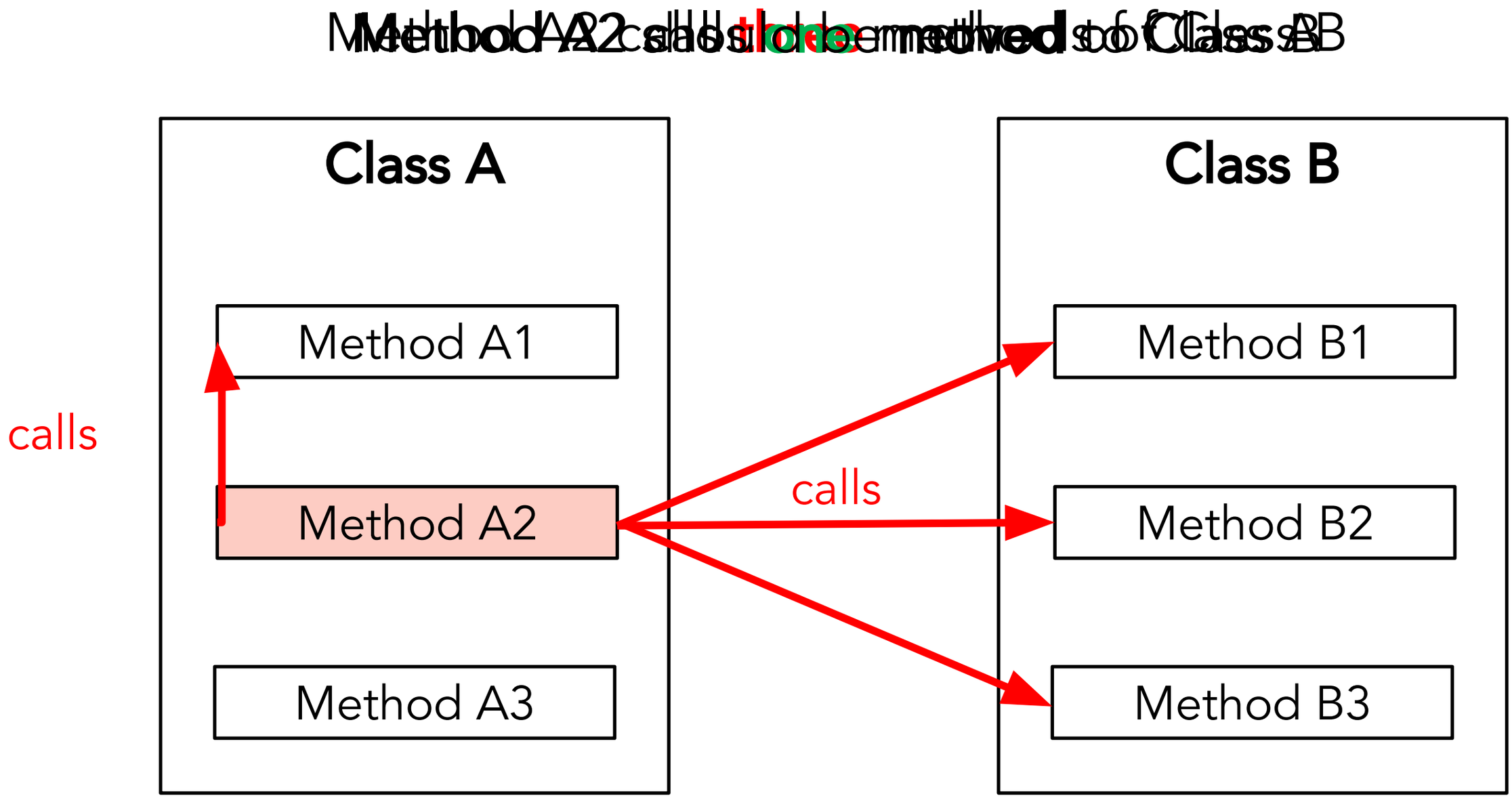
What is a code smell?

A symptom of maintenance problems in the source code

Different smell types help **identify** design problems

{oagbachi, emfernandes, afgarcia}@inf.puc-rio.br

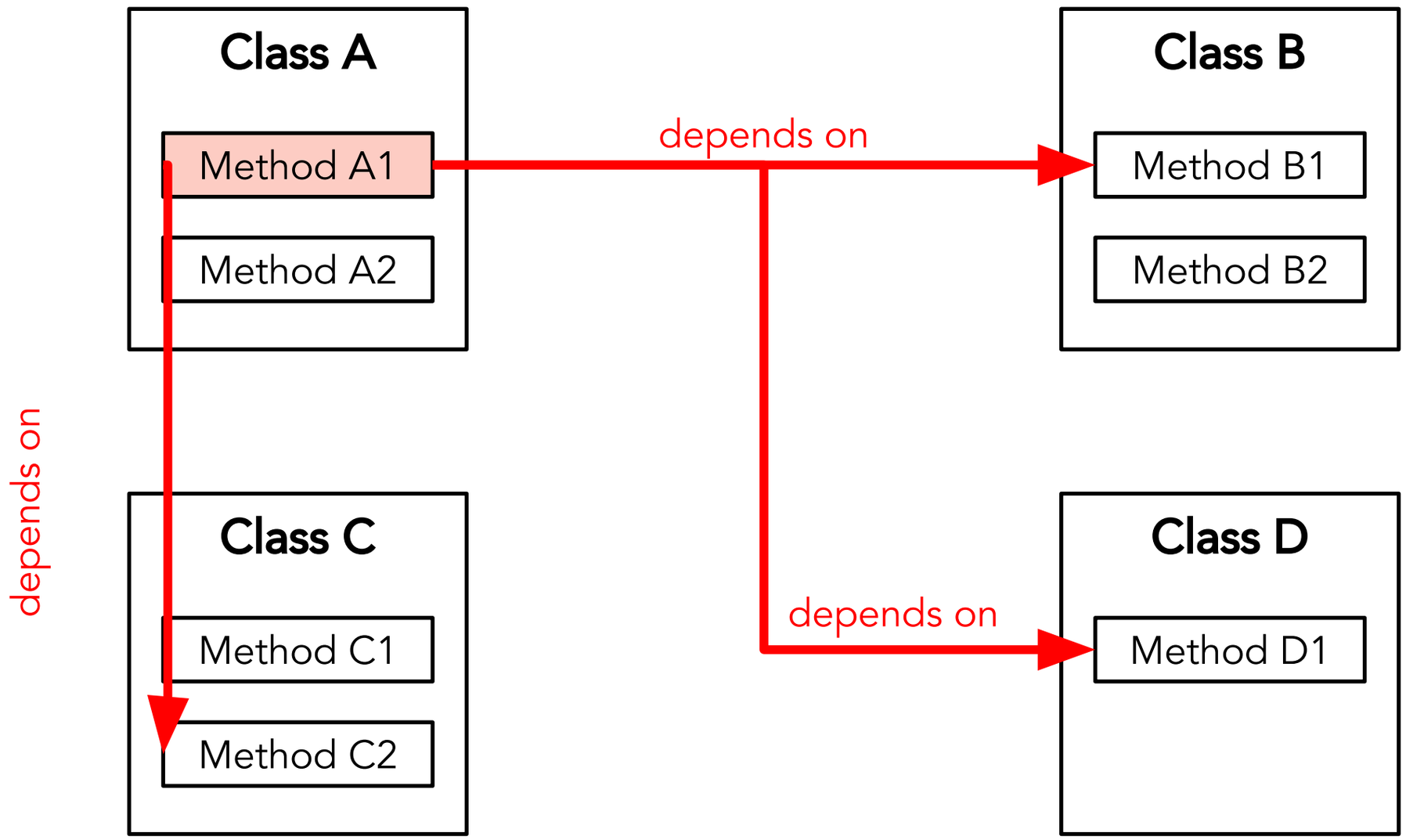
Example 1: Feature Envy



Feature Envy affects Method A2

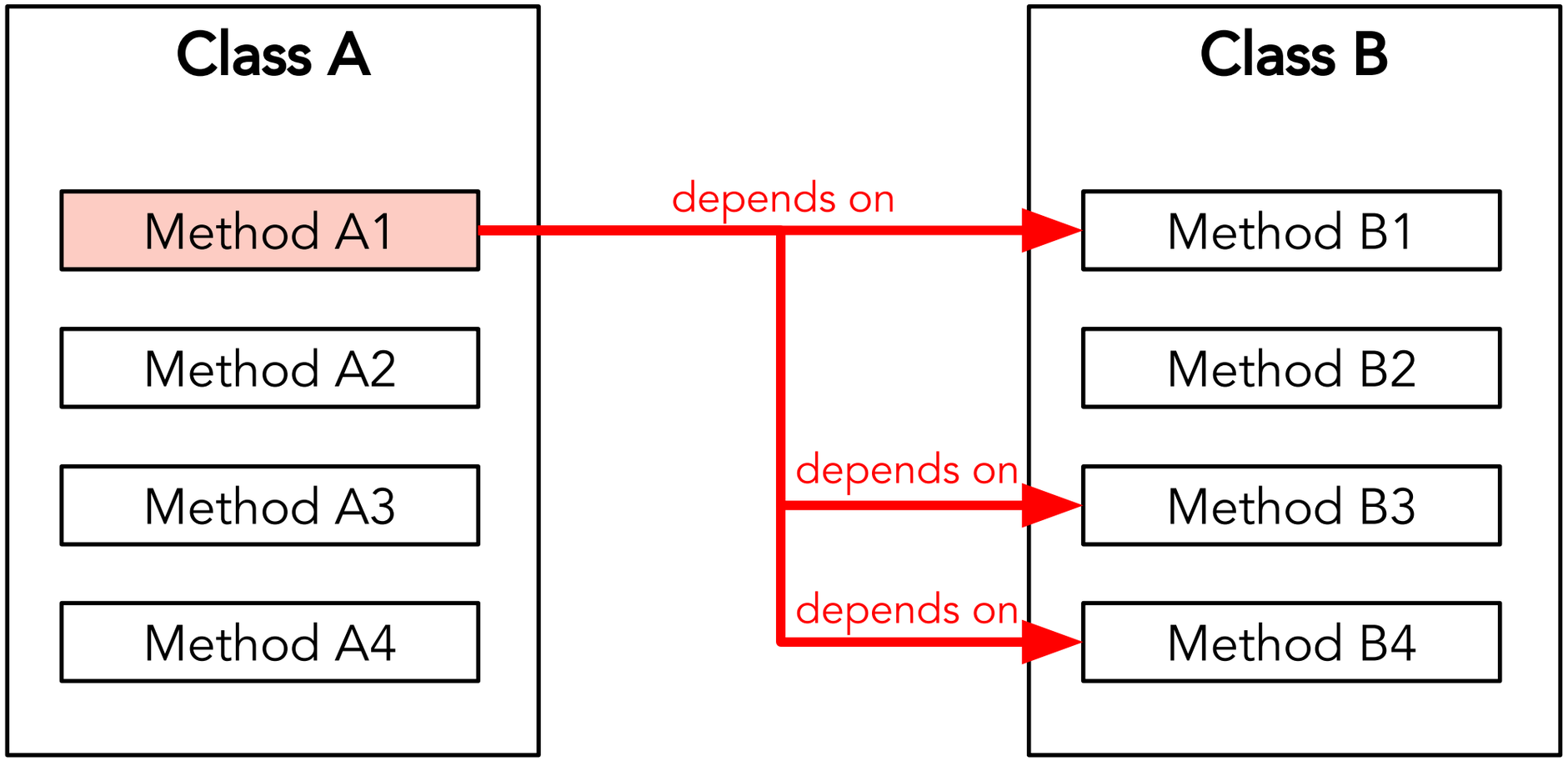
{oagbachi, emfernandes, afgarcia}@inf.puc-rio.br

Example 2: Dispersed Coupling



Dispersed Coupling affects Method A1

Example 3: Intensive Coupling



Intensive Coupling affects Method A1

{oagbachi, emfernandes, afgarcia}@inf.puc-rio.br

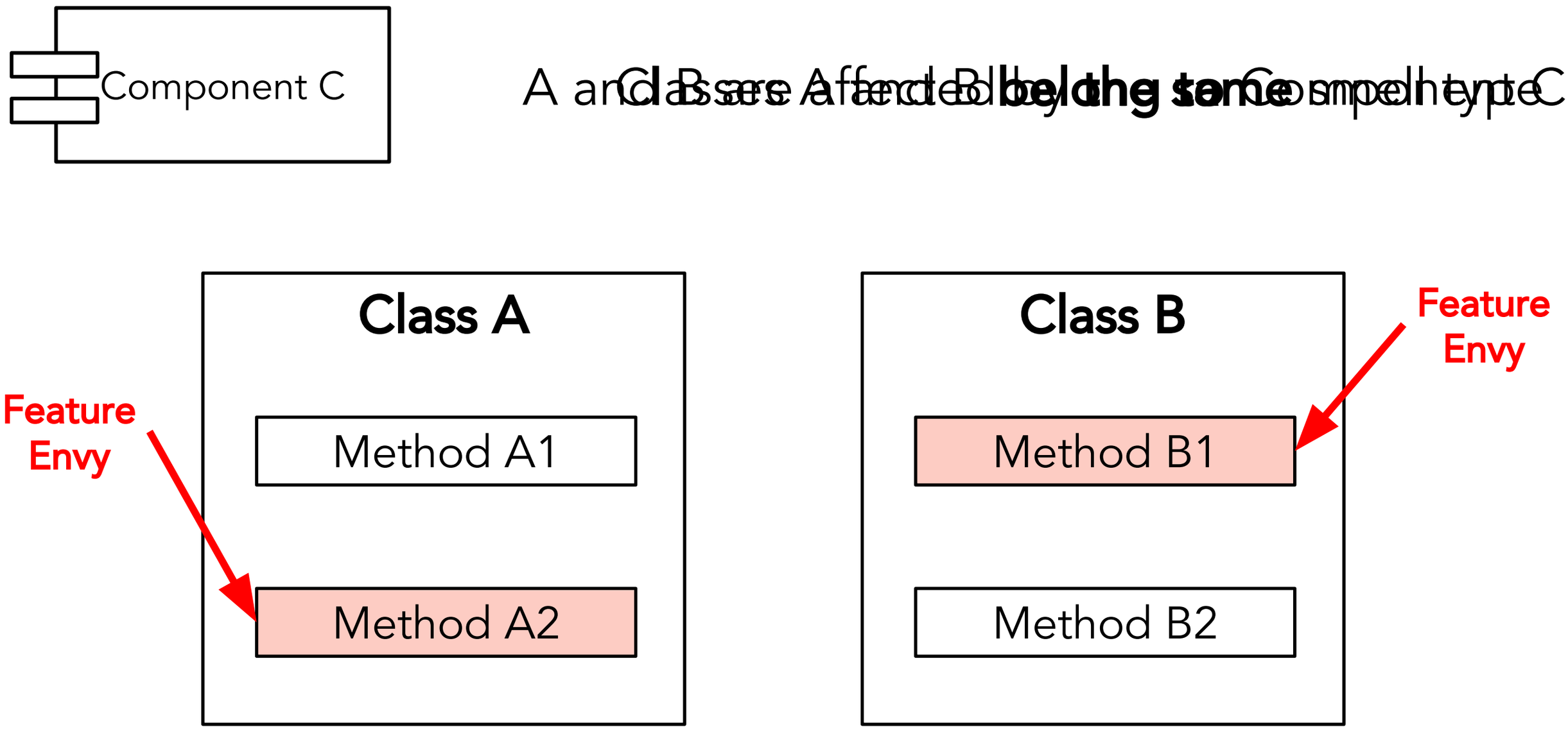
Code smell agglomeration

Group of code smells that **interrelate** in the source code

Relationships might be **explicit** (in the code) or **implicit** (concern)

{oagbachi, emfernandes, afgarcia}@inf.puc-rio.br

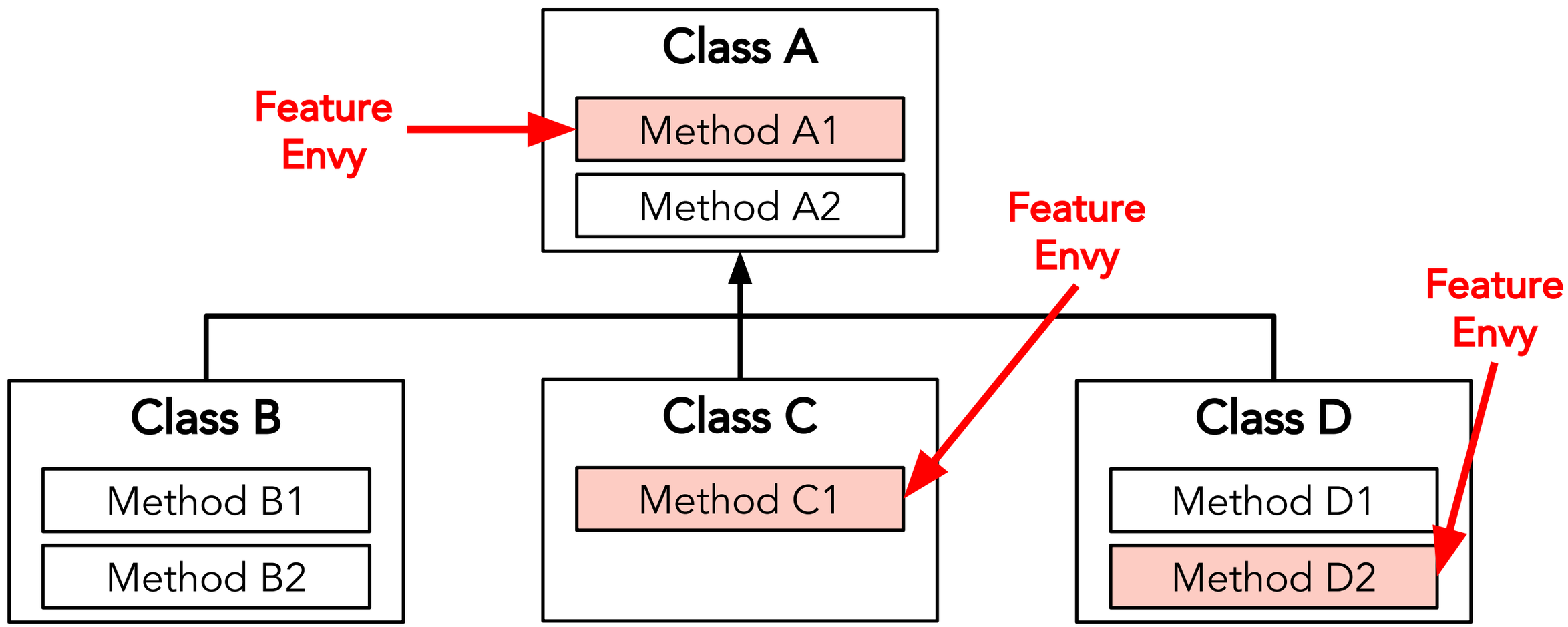
Example 1: Intra-component Agglomeration



Feature Envy instances affecting A2 and B1 form an agglomeration

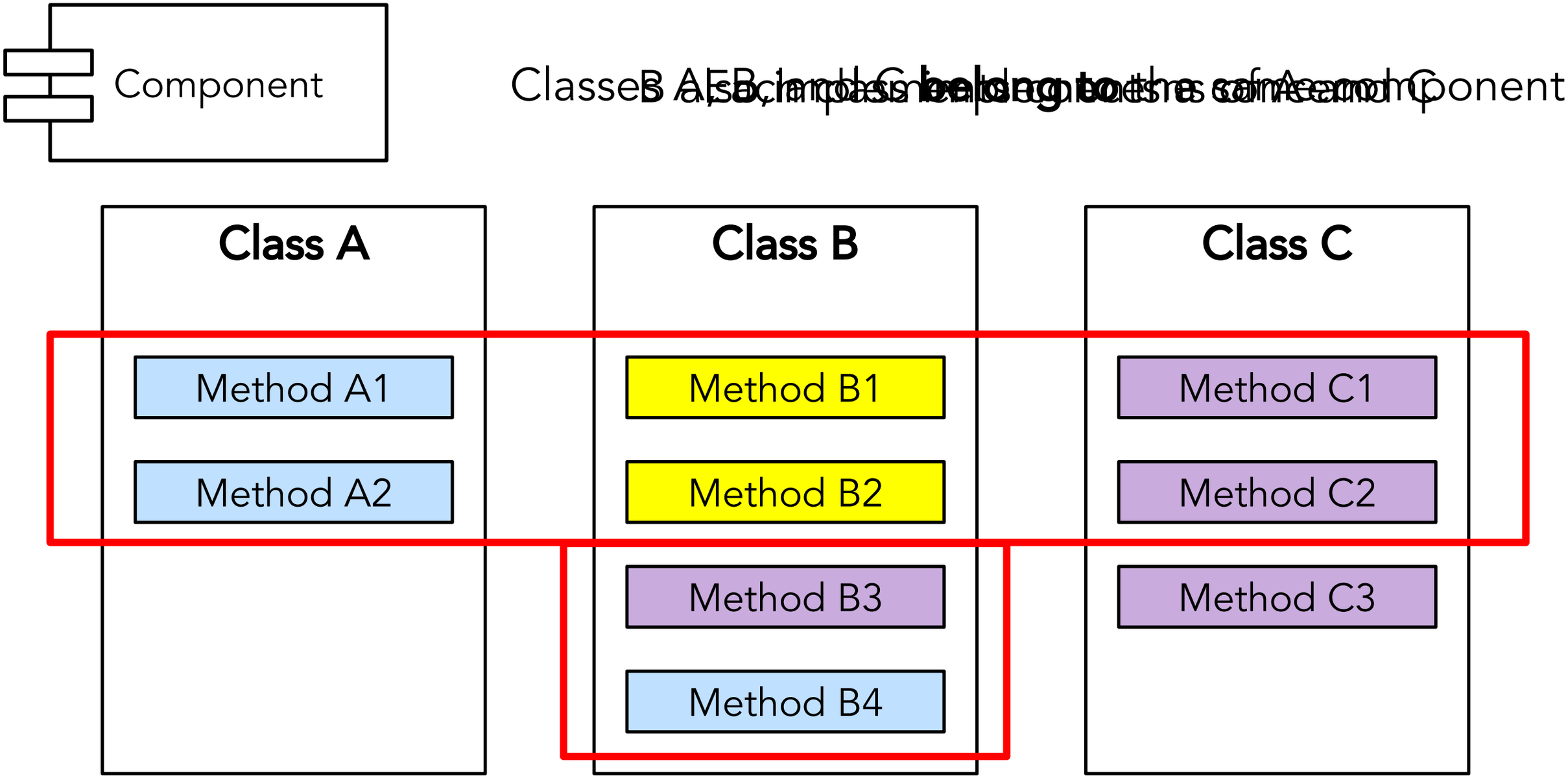
Example 2: Hierarchical Agglomeration

A, C, and D are affected by the same Feature Envy type



Feature Envy instances affecting A1, C1, and D2 form an agglomeration

Example 3: Concern Overload Agglomeration



Smells affecting B1, B2, B3, and B4 form an agglomeration

{oagbachi, emfernandes, afgarcia}@inf.puc-rio.br

The Opus Research Group

Identifying Design Problems with Code Smell Agglomerations

Basic Concepts

Benedicte Agbachi, Eduardo Fernandes, and Alessandro Garcia

OPUS Research Group

LES | DI | PUC-Rio - Brazil