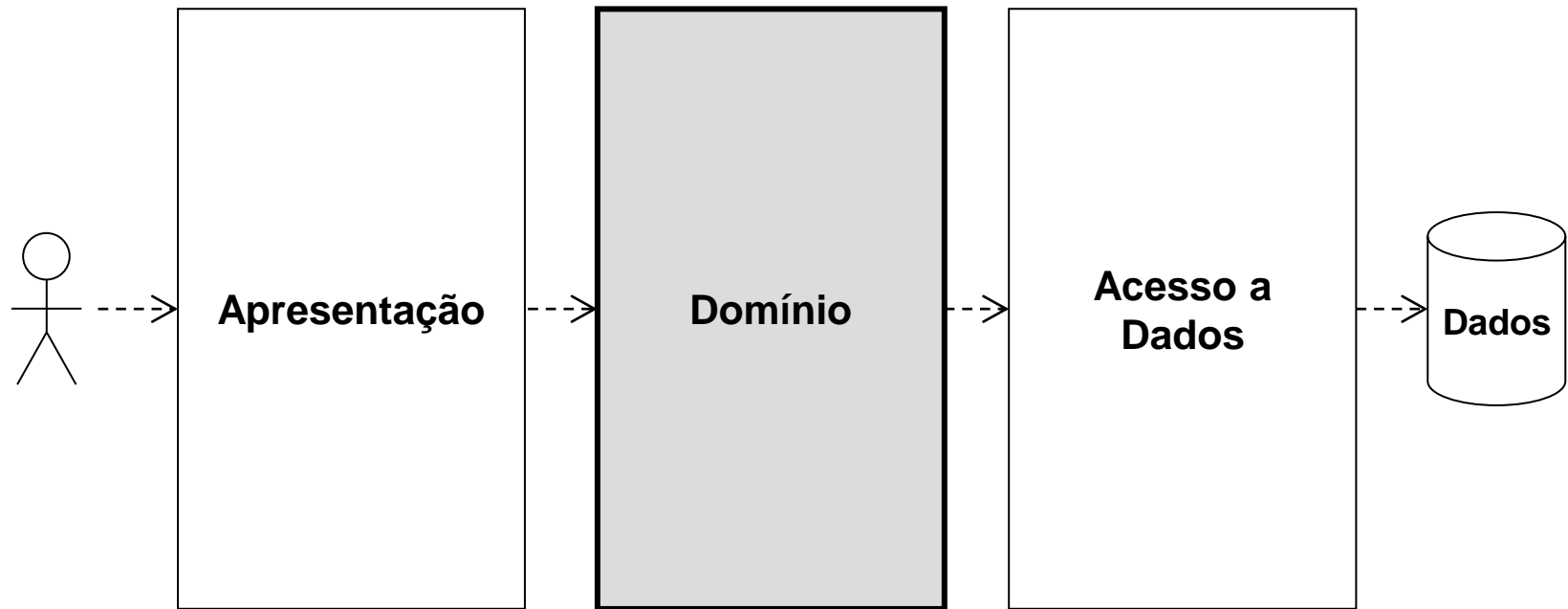

Engenharia de Software

Arquitectura Orientada ao Domínio

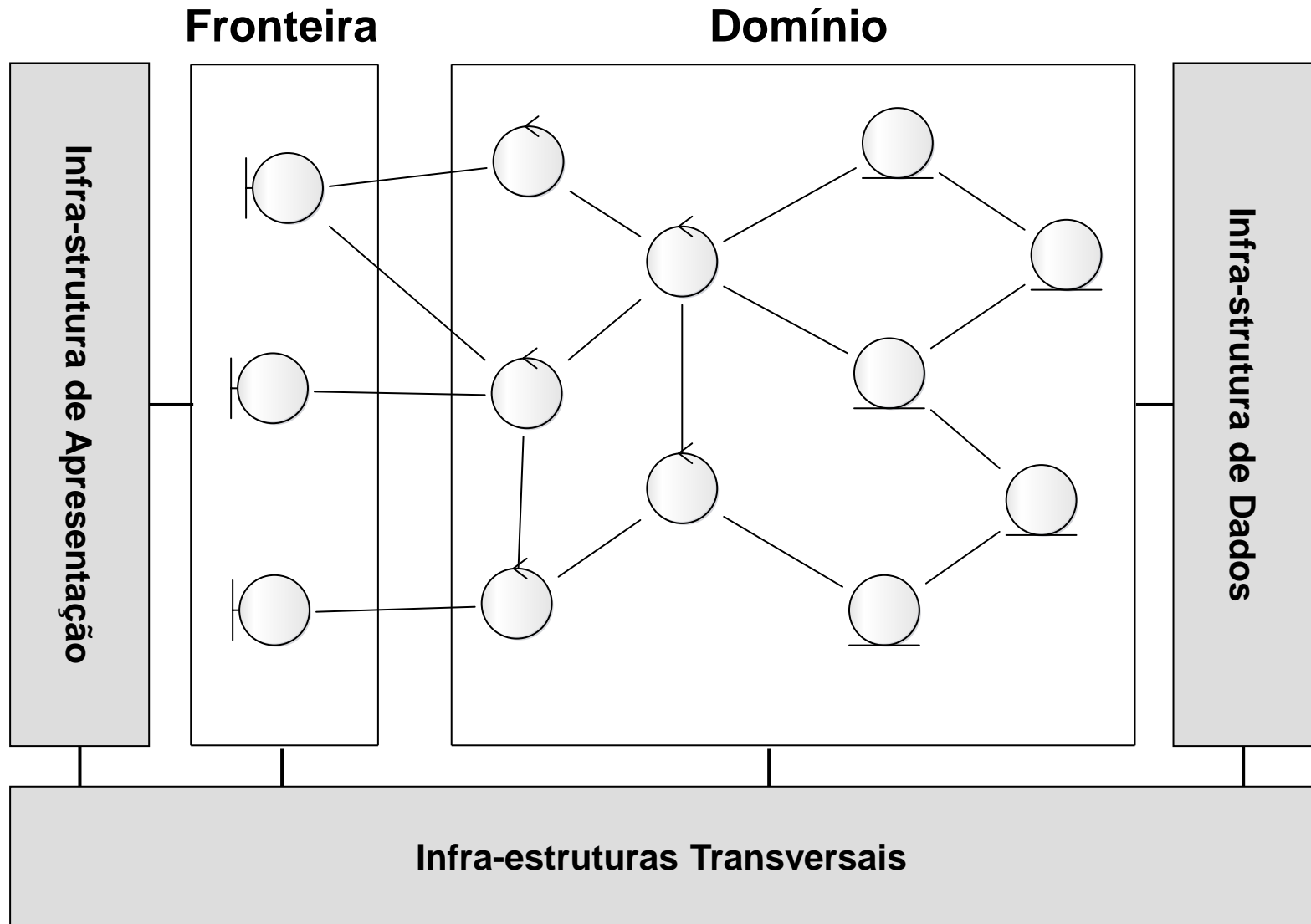
Luís Morgado

Instituto Superior de Engenharia de Lisboa
Departamento de Engenharia de Electrónica e Telecomunicações e de Computadores

Arquitetura de 3 Camadas



Arquitetura Orientada ao Domínio



Padrões de Domínio

- **Entidade**

- **Identidade** própria, que se mantém ao longo da operação do sistema
- Restrição de **unicidade**
- Manutenção de **estado** (*stateful*)

- **Objecto valor**

- Mantém atributos de um elemento de domínio
- Ausência de **identidade**
- Imutáveis
- Partilhados

Padrões de Domínio

- **Agregado**

- Agregação e contexto de entidades relacionadas
- Mantém a consistência de objectos relacionados
- Tem um **entidade raiz**
 - Identidade global
 - Objectos agregados têm identidade local
 - Do exterior apenas a raíz é referenciada
- Isola os elementos interiores do exterior
 - Encapsulamento

Padrões de Domínio

- **Fábrica**

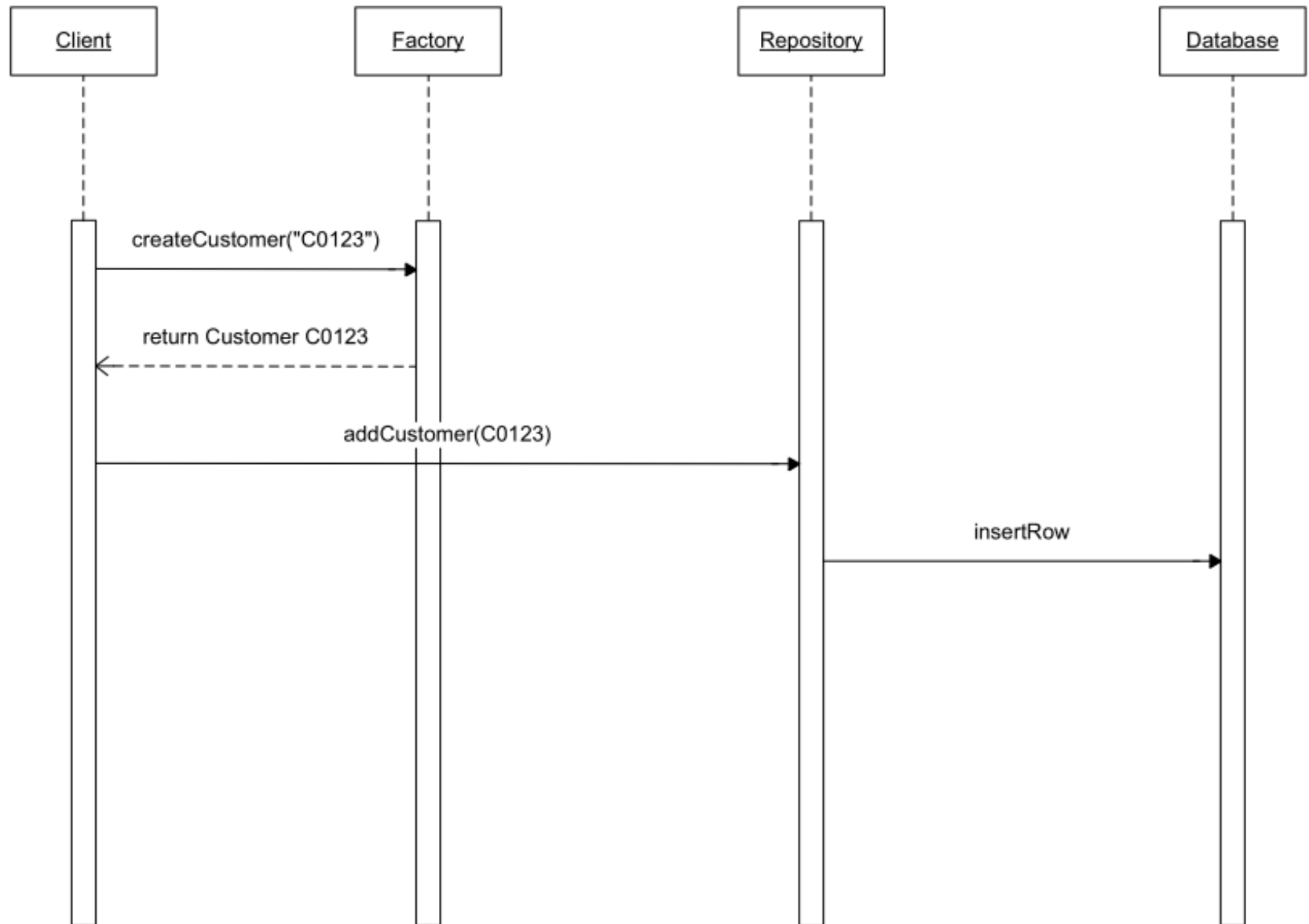
- **Gestão de criação** de objectos do domínio
 - Para objectos do domínio complexos
- Encapsulamento de conhecimento acerca do modo de criação dos objectos
 - Garantia de coesão e encapsulamente
 - Em especial no caso de **agregados**
- Processo de criação opera de forma atómica

Padrões de Domínio

- **Repositório**

- **Gestão de armazenamento** de objectos do domínio
- Encapsulamento do processo de armazenamento e obtenção de objectos do domínio
- Pode utilizar diferentes estratégias de armazenamento
- Funcionalidade
 - Inserir
 - Obter
 - Actualizar
 - Remover

Padrões de Domínio

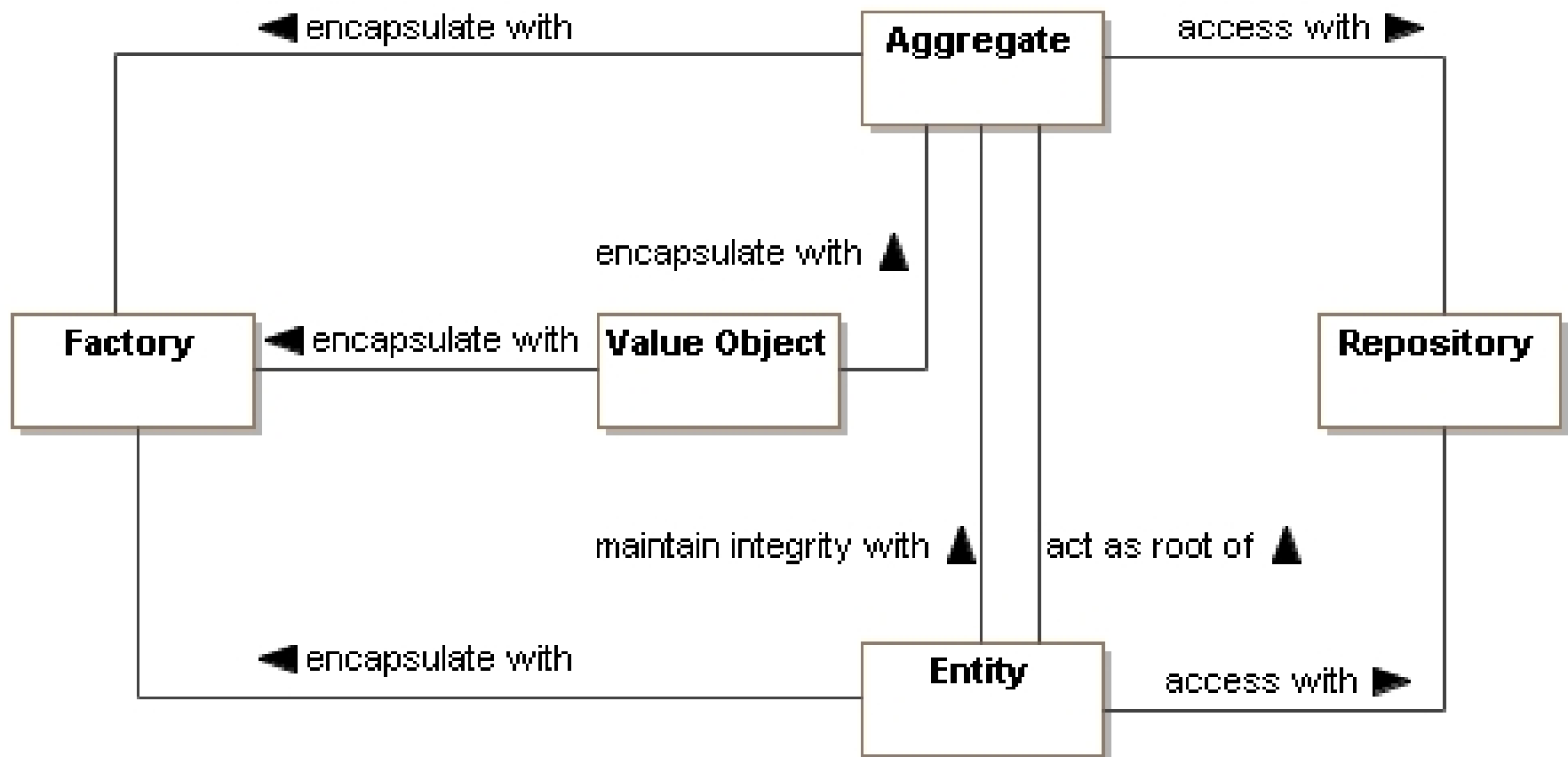


Padrões de Domínio

- **Serviço**

- Representa comportamento relevante do domínio que não é específico de um elemento de domínio
- **Coordena** a actividade de diferentes objectos
- Não mantém estado (***stateless***)
- Não deve substituir a funcionalidade específica de objectos do domínio, pelo que as operações realizadas
 - referem-se a conceitos de domínio que **não pertencem de forma natural a entidades ou objectos valor**
 - referem-se a outros objectos do domínio
 - **não mantêm estado** (*stateless*)
- **Diferente de serviços de isolamento do domínio**

Padrões de Domínio



Identidade de Entidades

Obtenção de identidade *posterior* à criação da entidade

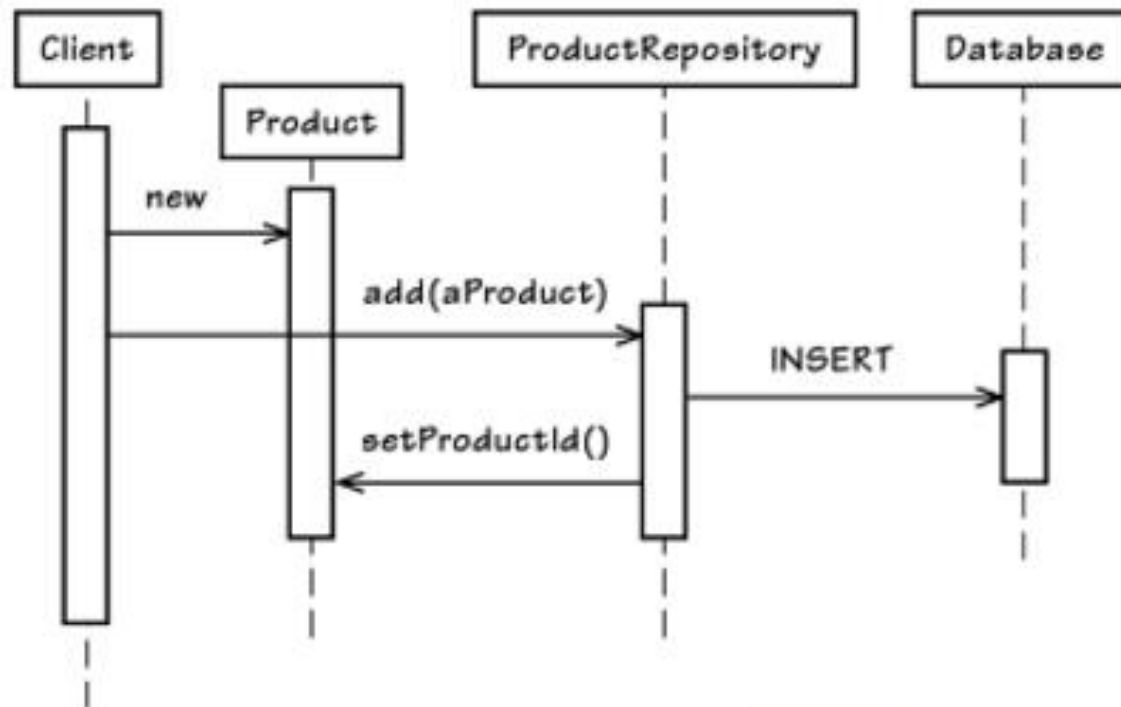


Figure 5.3 The simplest way to allocate a unique identity is to have the data store generate it the first time the object is persisted.

Identidade de Entidades

Obtenção de identidade *prévia* à criação da entidade

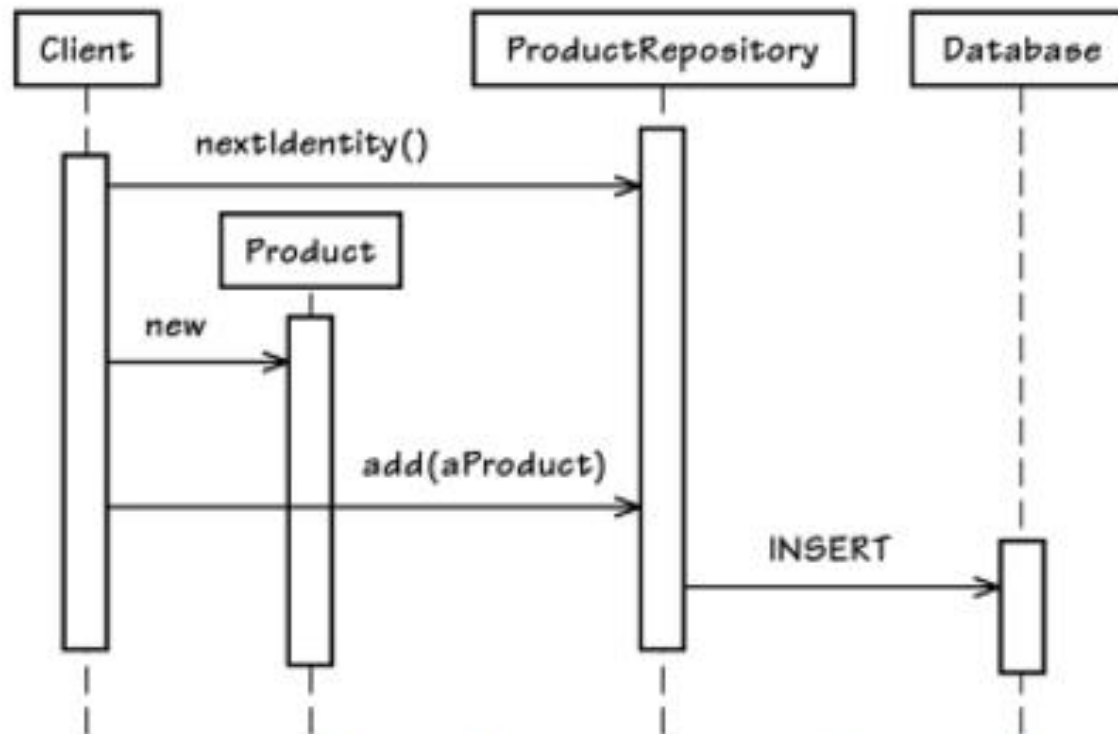
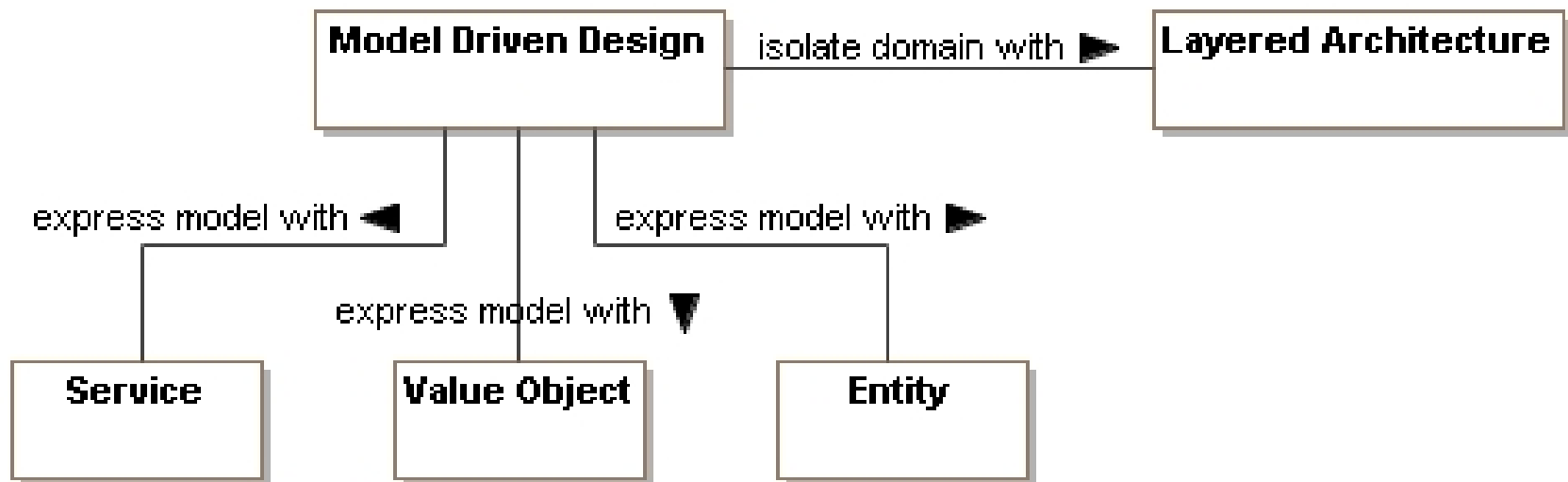


Figure 5.4 Here unique *identity* is queried from the *Repository* and assigned during instantiation. The complexities of *identity generation* are hidden behind the *Repository* implementation.

Padrões de Domínio

Arquitetura orientada ao domínio



Consequências ?

Bibliografia

[Pressman, 2003]

R. Pressman, *Software Engineering: a Practitioner's Approach*, McGraw-Hill, 2003.

[Gamma et al., 1995]

Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides, *Design Patterns: Elements of Reusable Object-Oriented Software*, Addison-Wesley, 1995.

[Shaw & Garlan, 1996]

M. Shaw, D. Garlan, *Software Architecture: Perspectives on an Emerging Discipline*, Prentice-Hall, 1996.

[Vernon, 2013]

V. Vernon, *Implementing Domain Driven Design*, Addison-Wesley, 2013.

[Parnas, 1972]

D. Parnas, *On the Criteria to Be Used in Decomposing Systems into Modules*, Communications of the ACM 15-12, 1968.

[Kruchten, 1995]

F. Kruchten, *Architectural Blueprints - The "4+1" View Model of Software Architecture*, IEEE Software, 12-6, 1995.

[Burbeck, 1992]

S. Burbeck; *Applications Programming in Smalltalk-80(TM): How to use Model-View-Controller (MVC)*, <http://st-www.cs.uiuc.edu/users/smarch/st-docs/mvc.html>, 1992

[Booch, 2004]

G. Booch, *Software Architecture*, IBM, 2004.