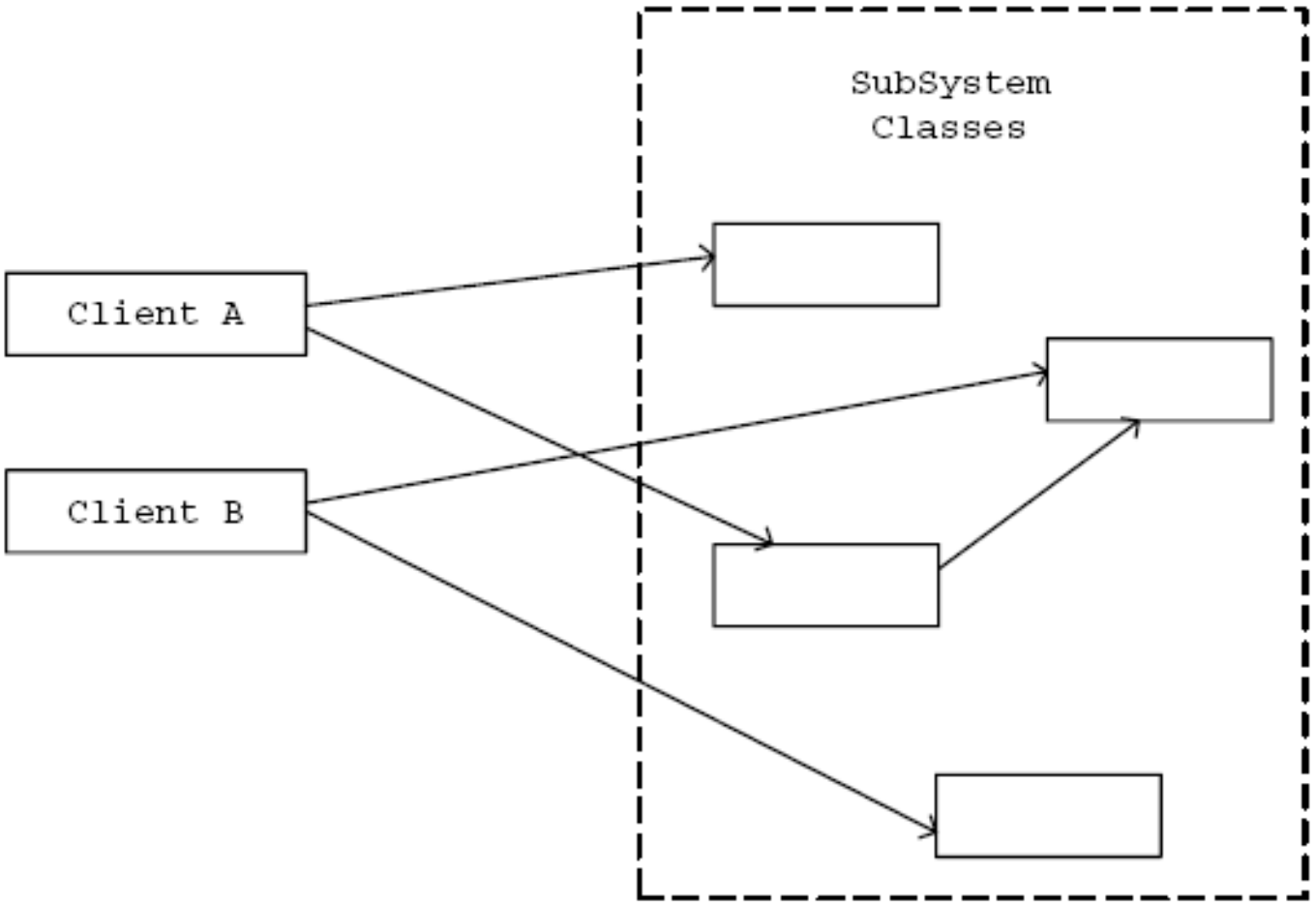
Façade

- Patró estructural
- “Agrupa” diverses classes que col·laboren entre sí per donar una una funcionalitat.
- Intermediari entre l'aplicació client i la funcionalitat *composta*
- NO afegeix noves funcionalitats.

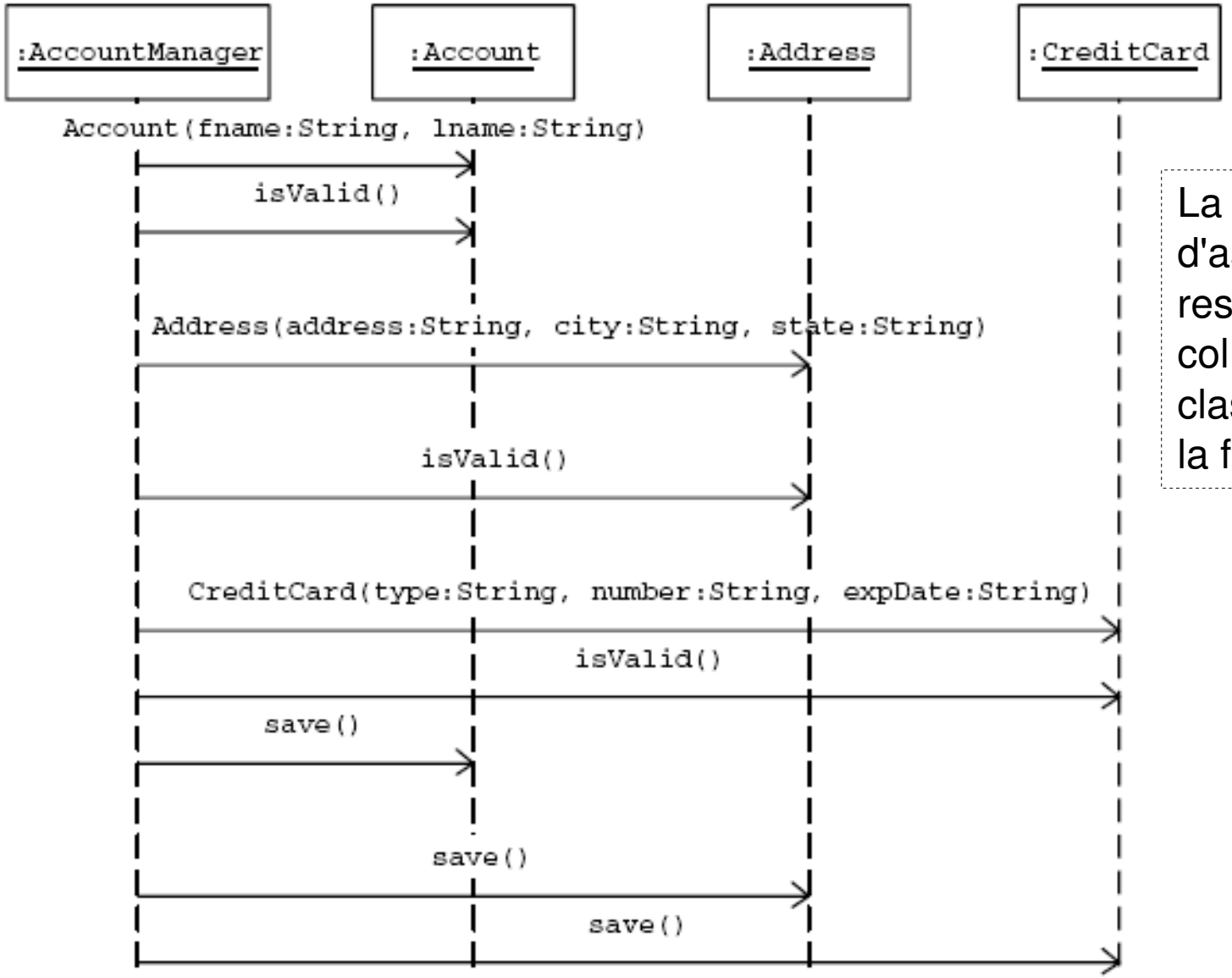
Objectius

- Amagar la complexitat dels subsistemes
- Restringir l'acoplament a un paquet o grup de classes.
- Amagar els subelements t'un element/classe que cal ser tractada com un tot

Subsistemes -> Complexitat

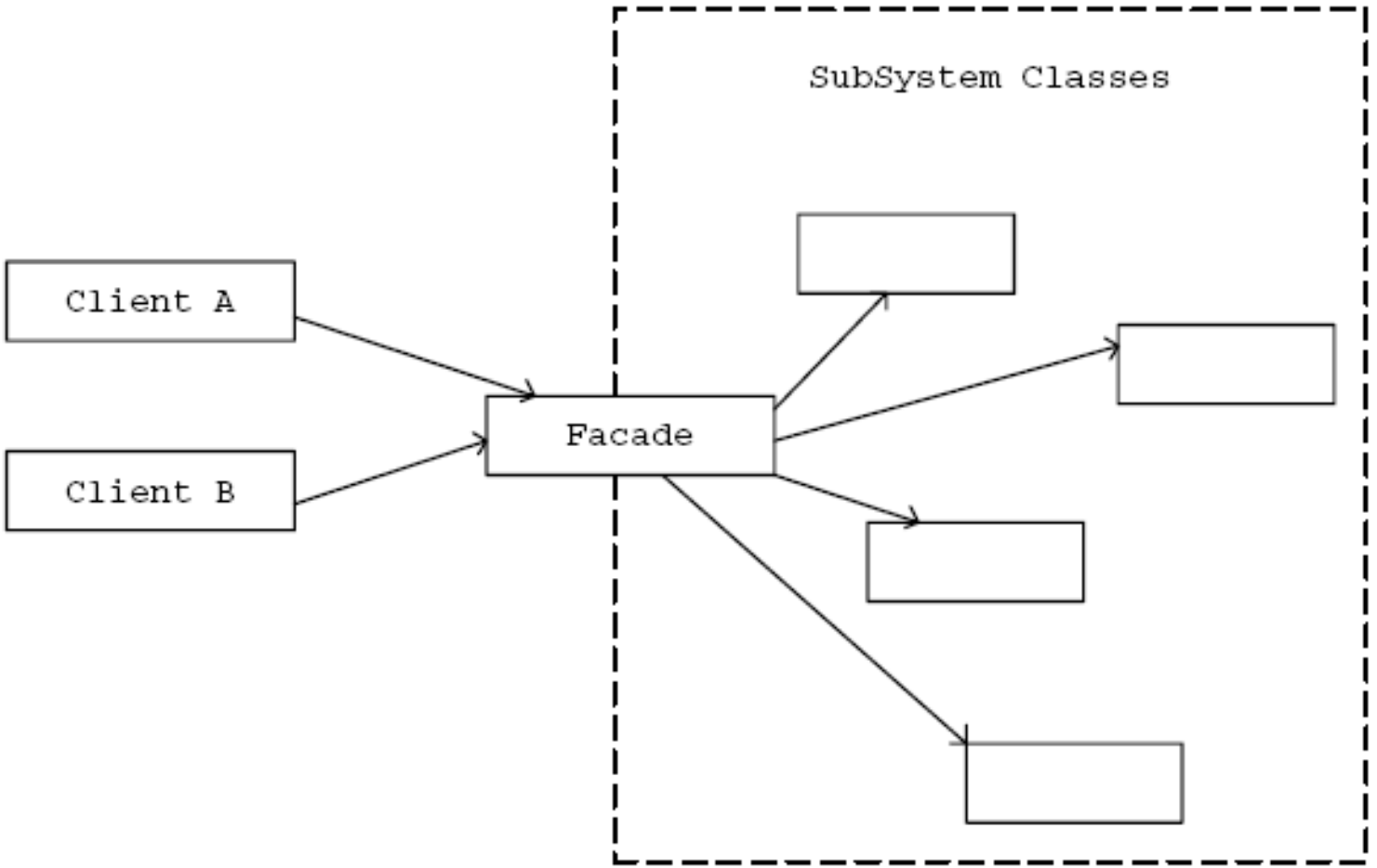


Subsistemes -> Complexitat



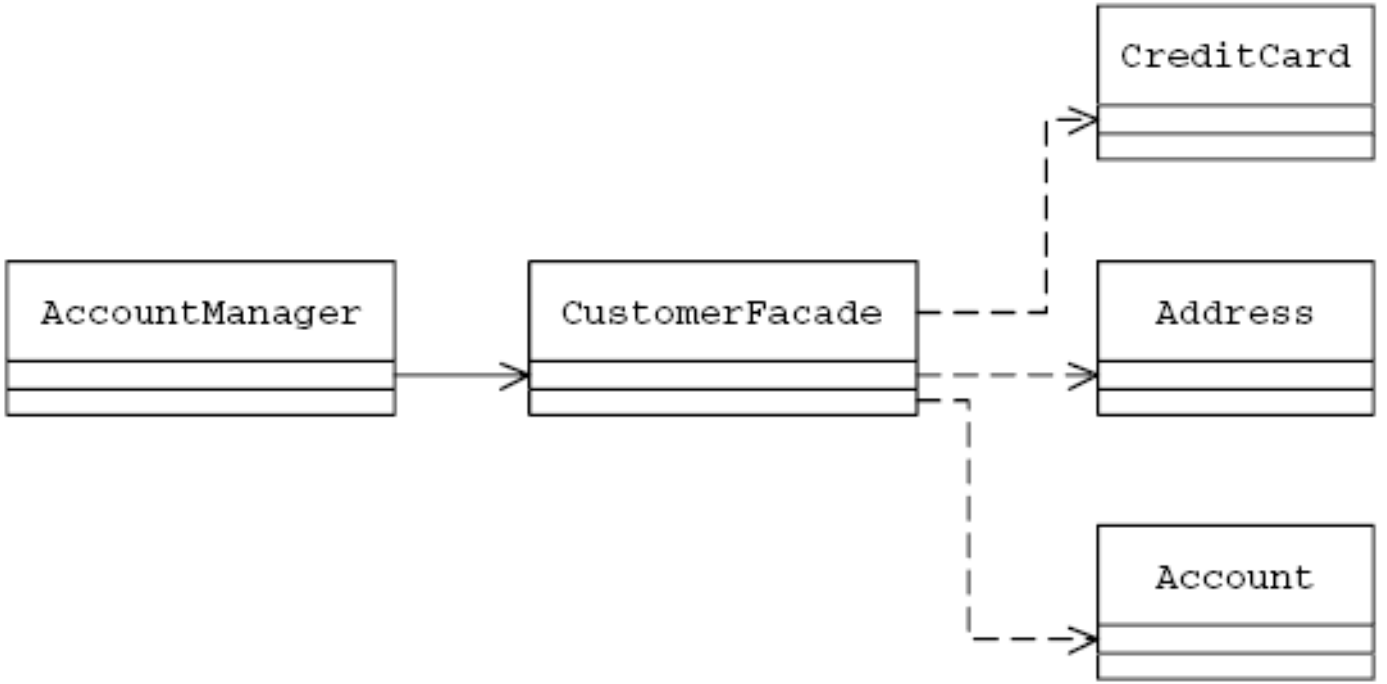
La classe client ha d'assumir totes les responsabilitats per fer col·laborar les diferents classes que li proporcionen la funcionalitat desitjada

Façade -> simplicitat

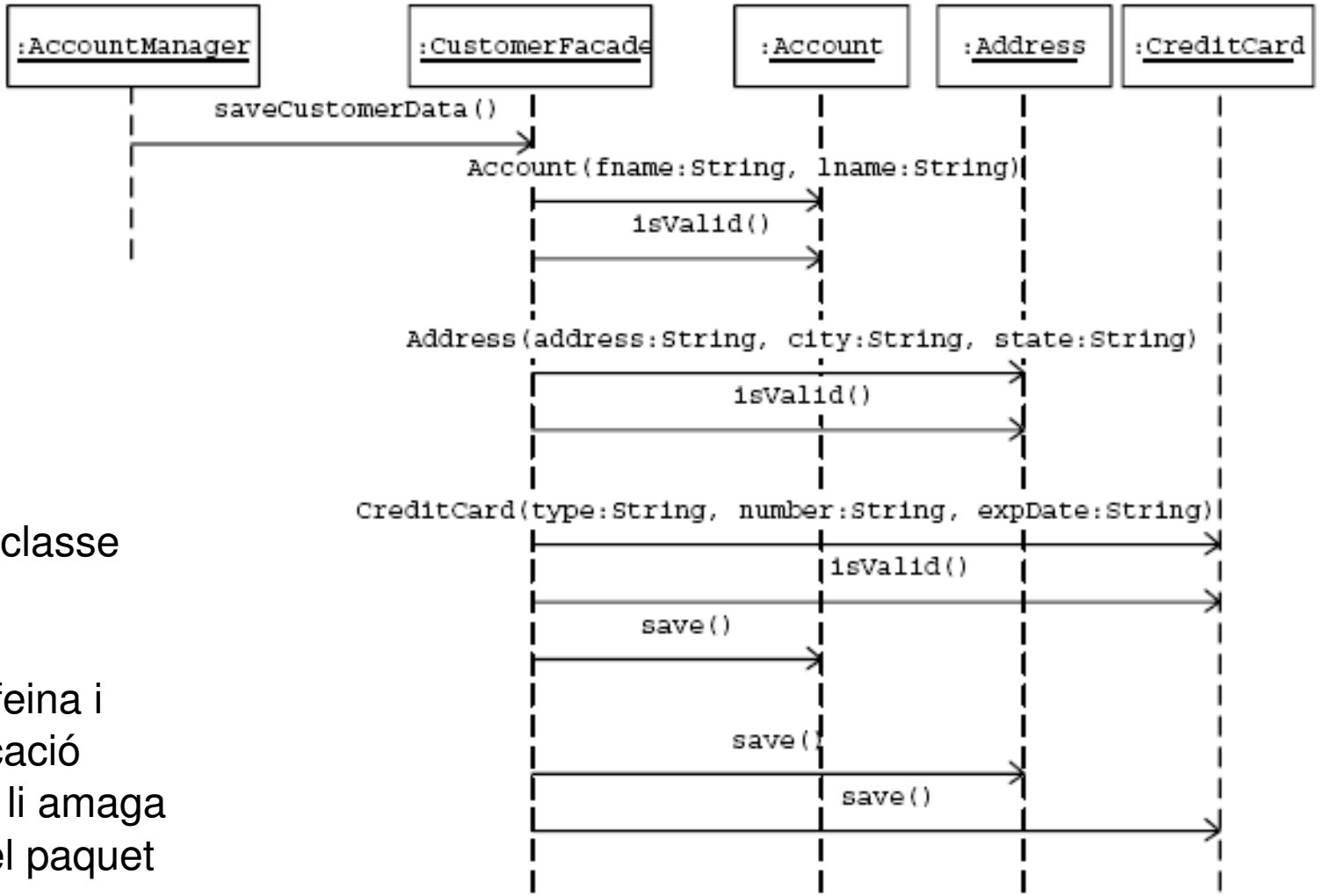


Façade -> Simplicitat

Apliquem el patró per crear una classe que faci d'intermediària i dugui a terme les tasques necessàries, l'aplicació client només parlarà amb aquesta classe



Façade -> Simplicitat



Façade és ara la classe que assumeix la responsabilitat descarregant de feina i complexitat l'aplicació client, alhora que li amaga les interioritats del paquet

Usos habituals

- Sol ser el 'cap visible' d'un paquet
- Englobar la funcionalitat d'un cas d'ús
- Es sol implementar junt amb el patró Singleton, sobretot per entitat globals que representen tot el sistema, l'entorn de l'aplicació o l'accés a la persistència (Façana de persistència).

Façade VS altres patrons

- Façade VS Adapter:
  - Adapter varia l'interfície d'una classe
  - Façade esdevé l'interfície d'un conjunt de classes
- Façade VS Decorator:
  - Decorator va embolcallant una vegada i una altre una classe per afegir-hi funcionalitats.
  - Façade embolcalla (una sola vegada) un conjunt de classes.