# Software Design and Architecture

## Architectural Views

*Agenda*

- goals of representing architecture
- more UML diagrams
  - sequence diagram
  - communication diagram
  - state machine diagram
  - activity diagram

Winter 2026. Instructor: Pengyu Nie.

Acknowledgements: slides adapted from previous versions by Mei Nagappan and Shane McIntosh, which are adapted from previous versions by Zhen Ming Jiang, Ahmed E. Hassan, Reid Holmes.

# Representing Software Architecture

- Software architecture is fundamentally about facilitating technical communication between project stakeholders
- Properties of representations:
  - Ambiguity: open to more than one interpretation?
  - Accuracy: correct within tolerances
  - Precision: consistent but not necessarily correct

# Architectural Views

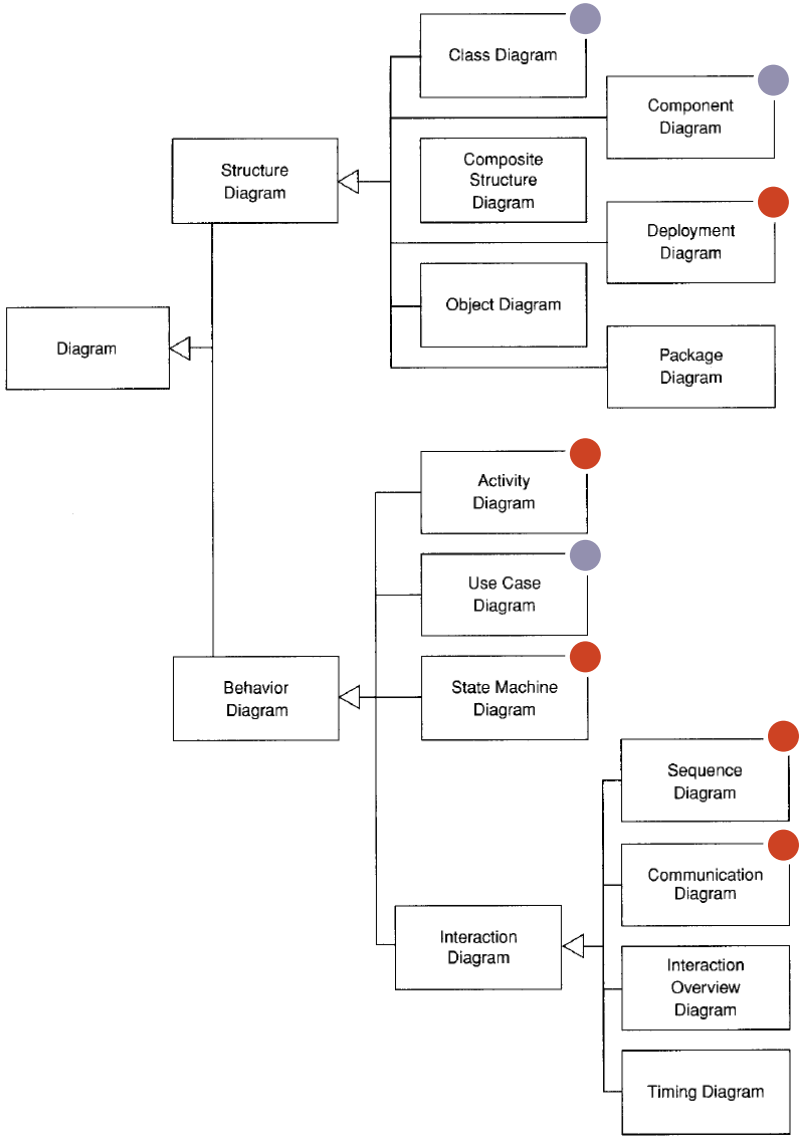
- Architectural views (UML diagrams)
  - different views focus on specific subsets of elements or subsets of relationships
  - views often focus on specific concerns or scenarios within a system
  - but… views can overlap; maintaining consistency between views is challenging

classification of UML diagram types overall two classes:

- for static structure
- for dynamic behavior

- ~~component diagram~~
- ~~class diagram~~
- deployment diagram

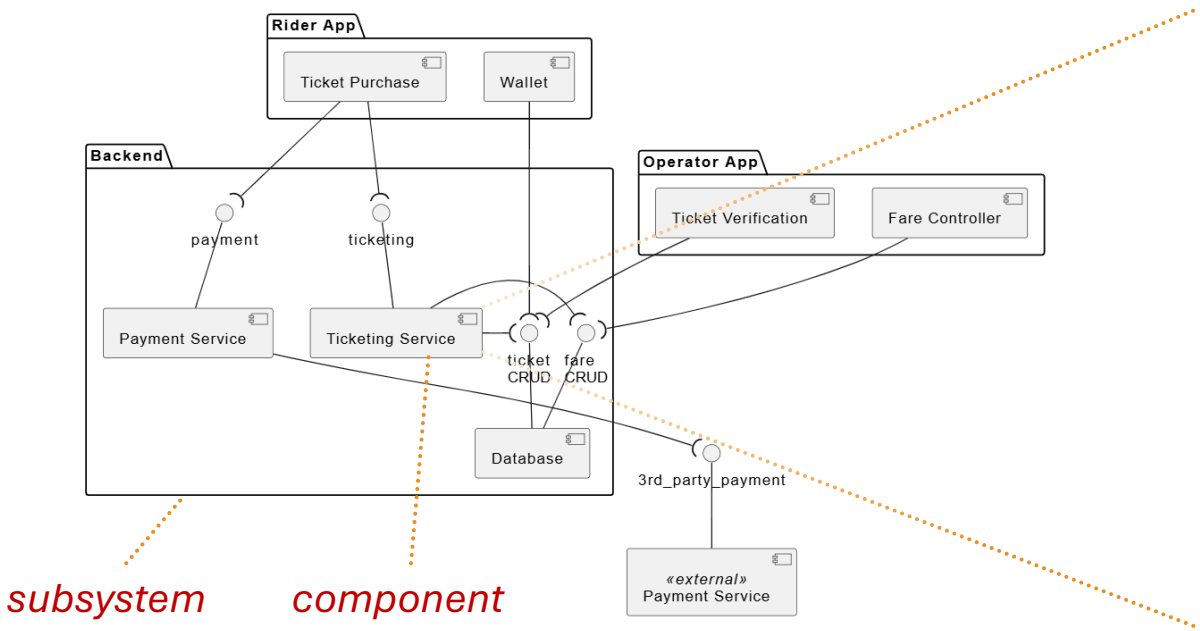
- ~~use case diagram~~
- sequence diagram
- communication diagram
- state machine diagram
- activity diagram

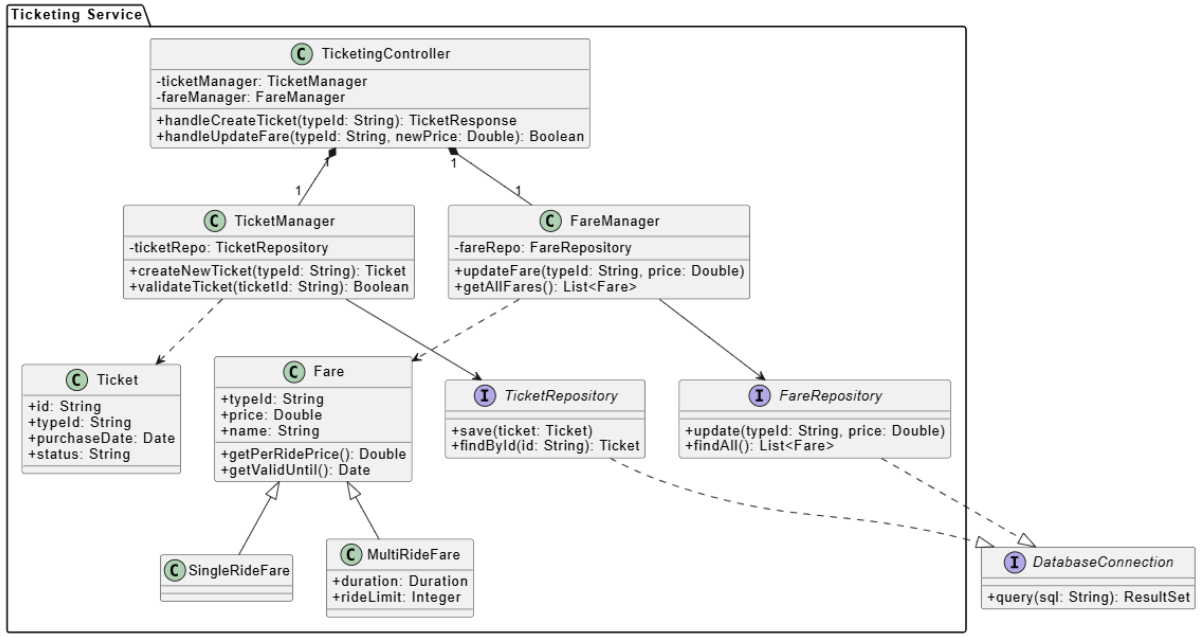


# Component Diagram & Class Diagram

structure high-level

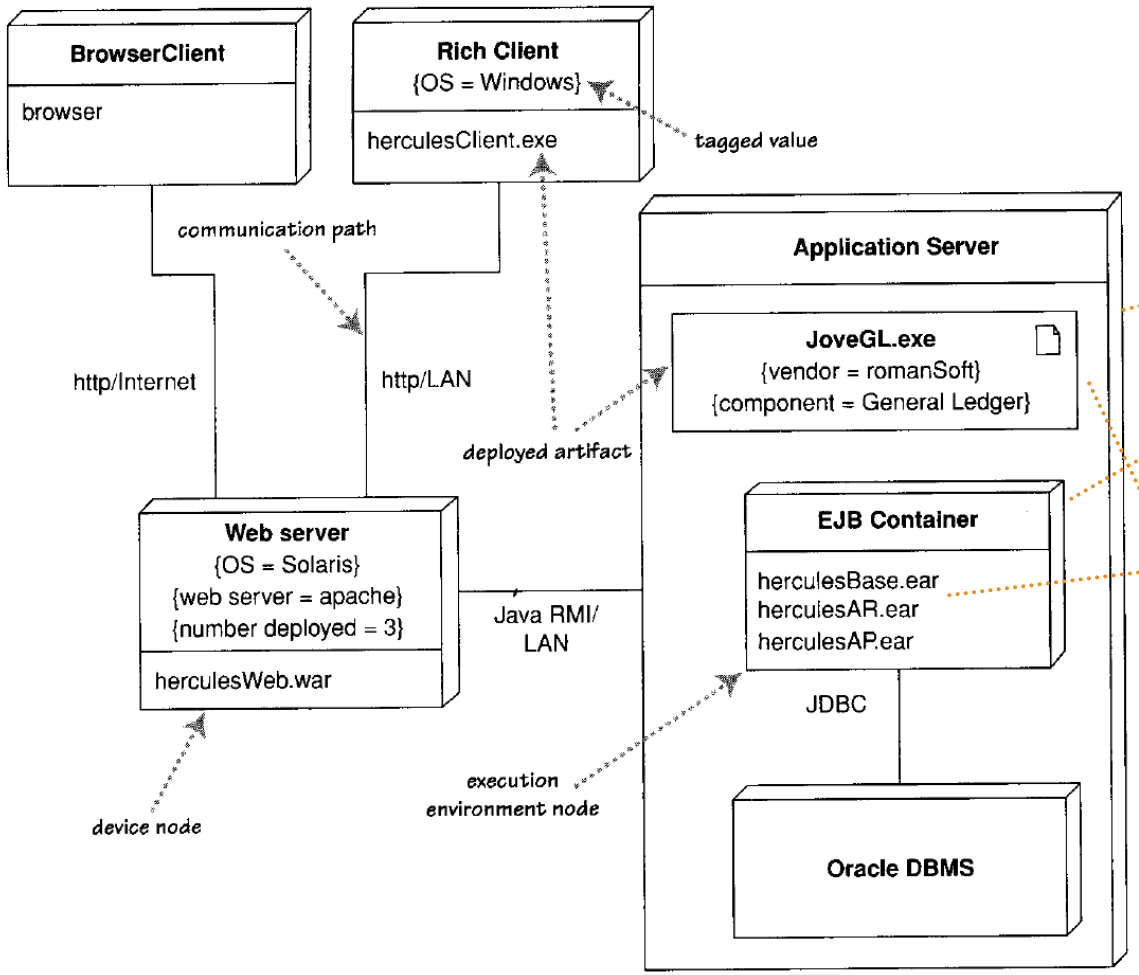
structure low-level



# Deployment Diagram

structure | physical




*node*

something that can host some software

can be device or execution environment

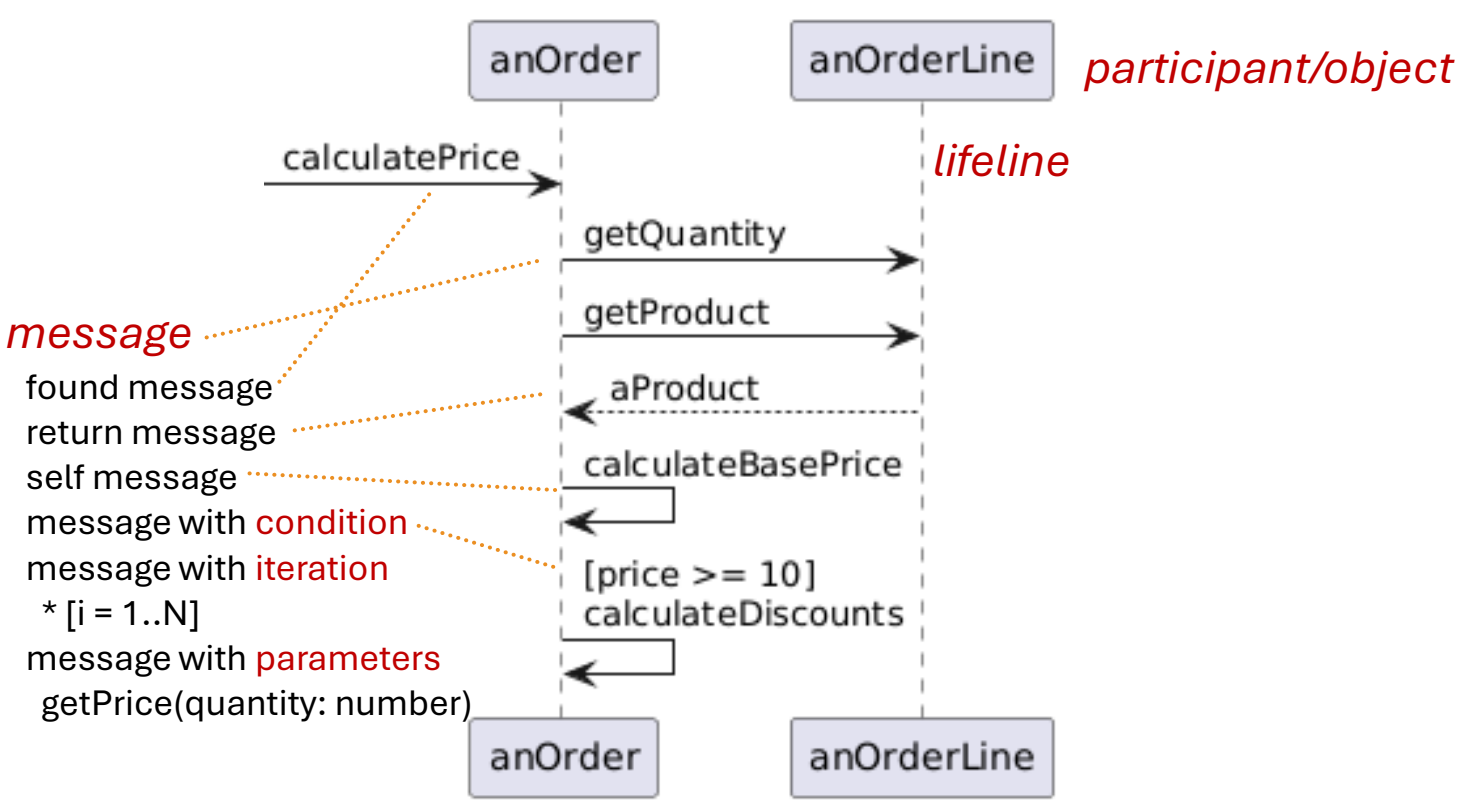
*artifact*

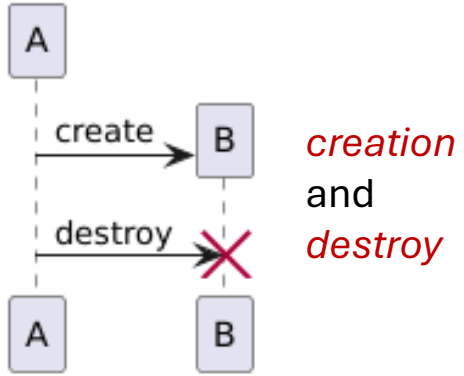
physical manifestations of software

executables, data files, configuration files, etc.

# Sequence Diagram

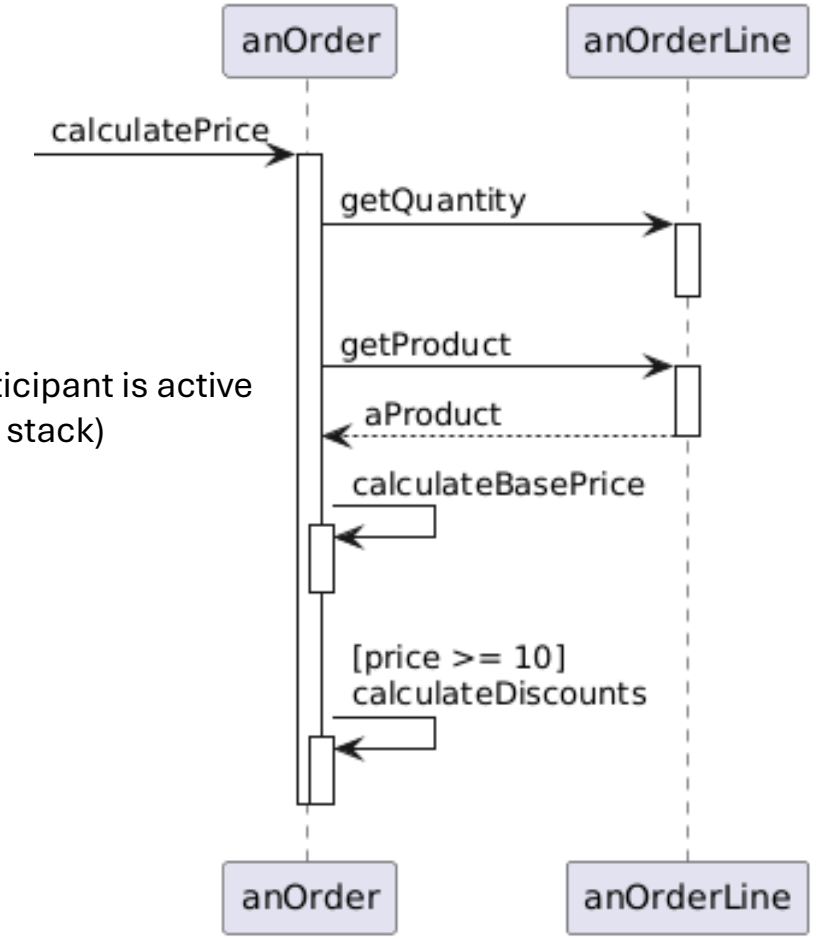
behavior | chronology



# Sequence Diagram

behavior | chronology

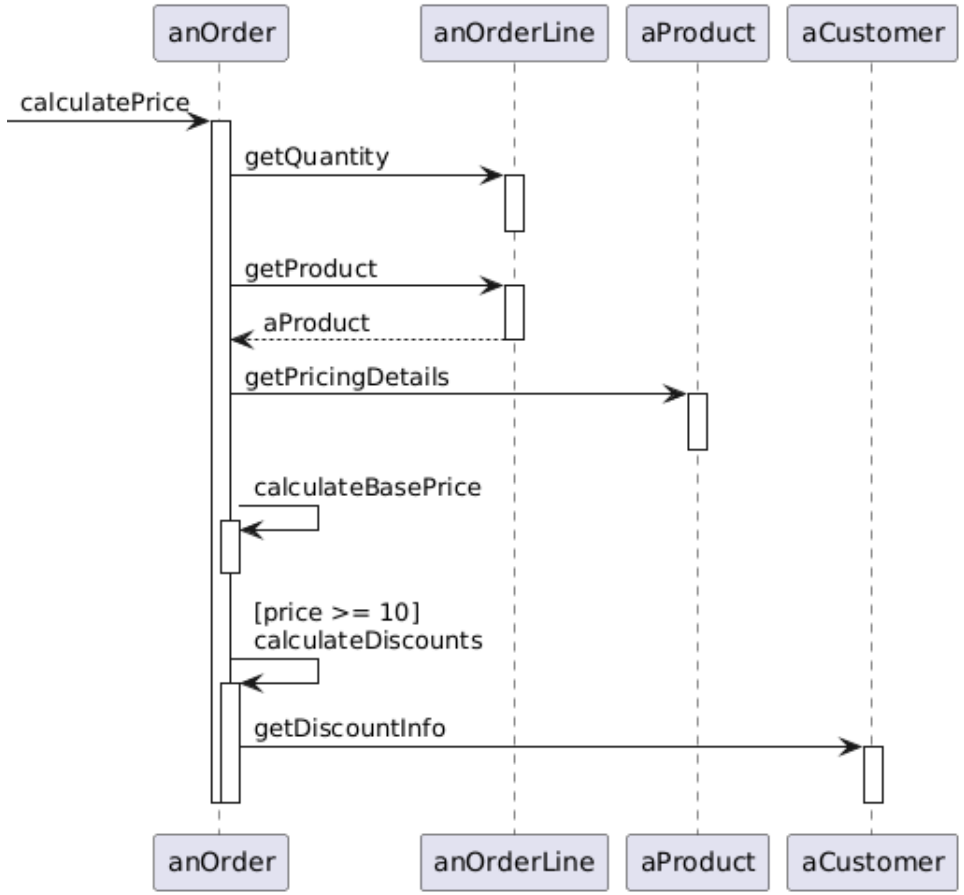



*activation bar*

shows when each participant is active
(the function call is on stack)

# Sequence Diagram

behavior | chronology



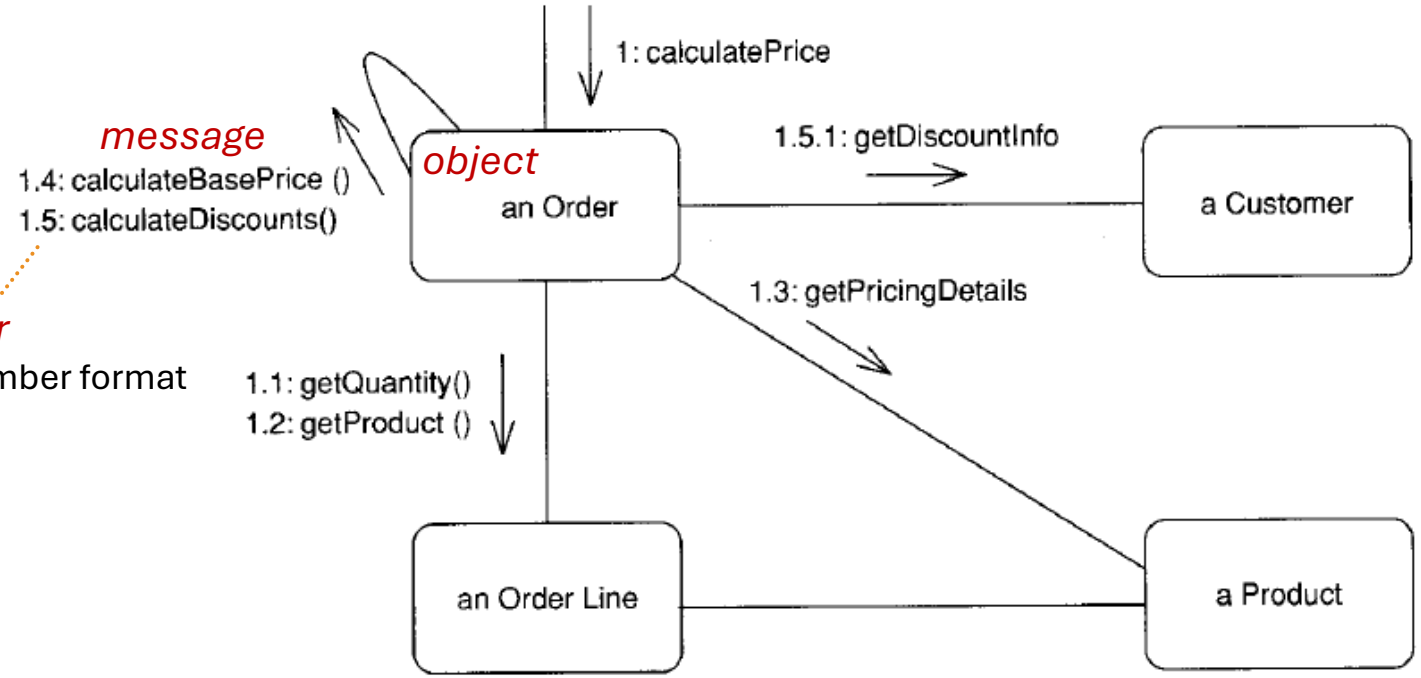
# Practice: sequence diagram for login

- Task: draw UML sequence diagram(s) for the user authentication of an app, as we add more requirements…
  - 1 (basic): User can register for a new account with username and password
  - 2 (basic): User can login with the correct username and password
  - 3: User is automatically logged-in after registration
  - 4: User can authenticate with a Google account (“Login with Google”)

# Communication Diagram

behavior | topology

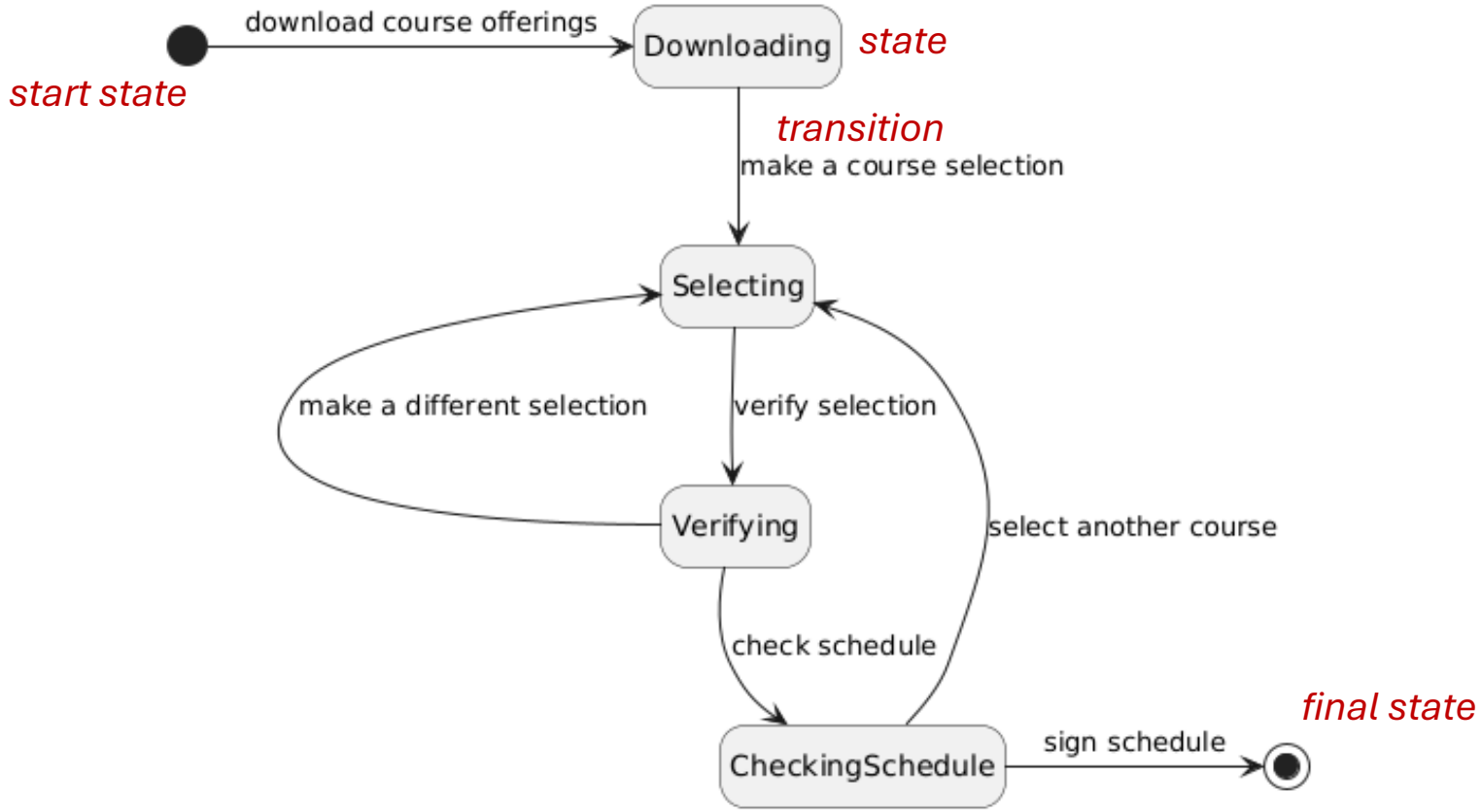
- alternative name: collaboration diagram (in UML v1)



# State Machine Diagram

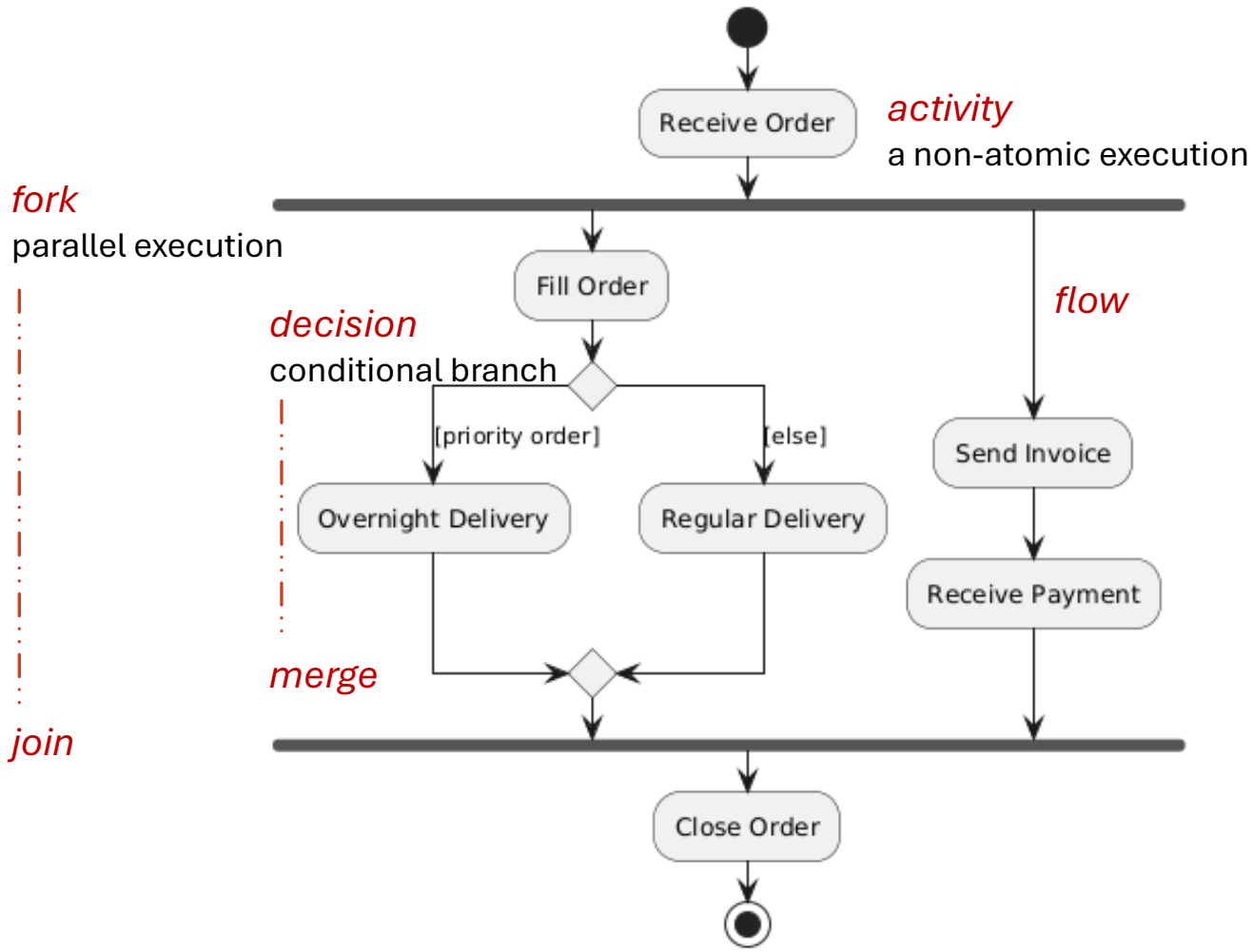
behavior | lifecycle

- alternative names: state diagram, state machine



# Activity Diagram

behavior | control flow



# Prescriptive vs. Descriptive Representations

- Prescriptive architecture dictates how the system will be built a priori (as conceived)
- Descriptive architecture captures how the system was actually built a posteriori (as implemented)

- Architectural degradation
  - drift: changes that are not captured in the current architecture but do not violate it
  - erosion: changes that violate the current architecture