Course Introduction

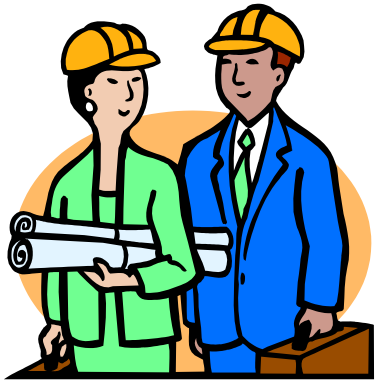
- Introduction
- Basics
- Structural Modeling
- Behavioral Modeling

UML Today

- **Current – 2.4.1**
  - August 2011
- **Upcoming – 2.5**
  - Beta, October 2012

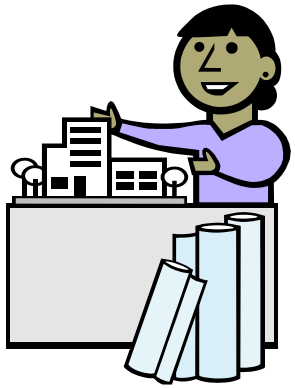
Communication Problems

Common Users of UML Diagrams



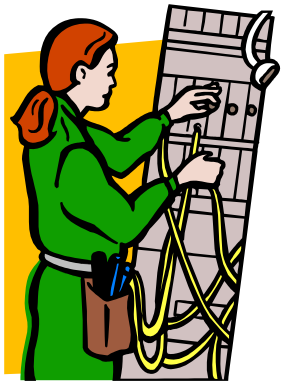
Product Owner

Business Analyst

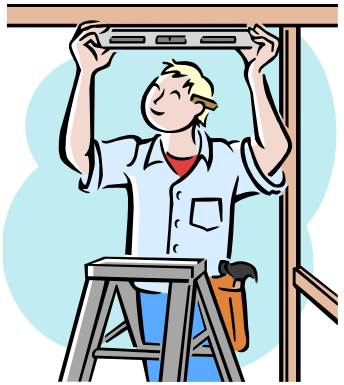


Architect

Operations



Quality Assurance



Developer

Same Thing – Different Language

Suunnittelen ohjelmisto

Ik ontwerp software

मैं सॉफ्टवेयर डिजाइन

Progettazione software

Ich habe design software

I design software

소프트웨어 디자인

Дизайн софтуер

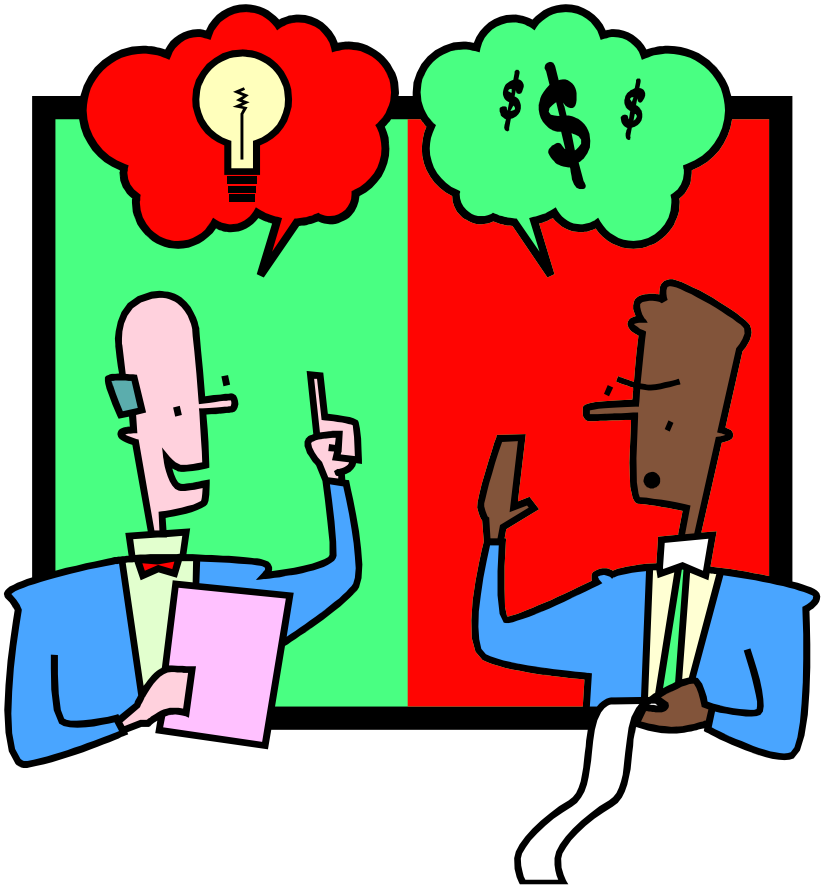
Tôi thiết kế phần mềm

Je conçois des logiciels

Diseño software

Я розробки програмного забезпечення

Business?



Technology

Smalltalk

Lisp

Python

C/C++

XSLT

Prolog

DML

F#

JavaScript

C#

DDL

Scala

SOA

SQL

Java

Pascal

TypeScript

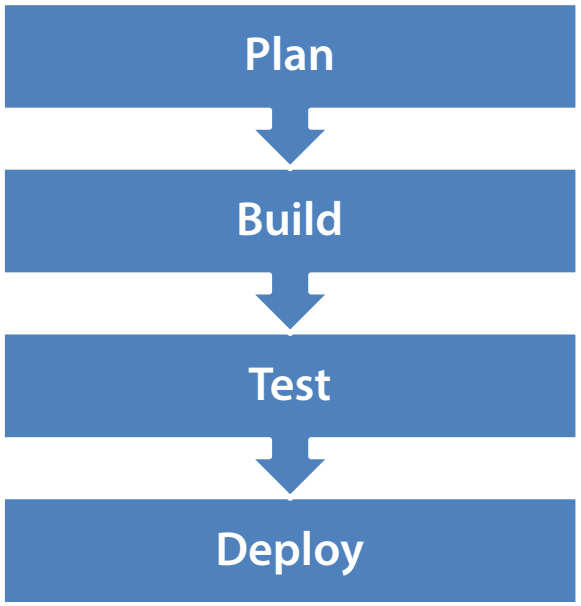
VB.NET

Cobol

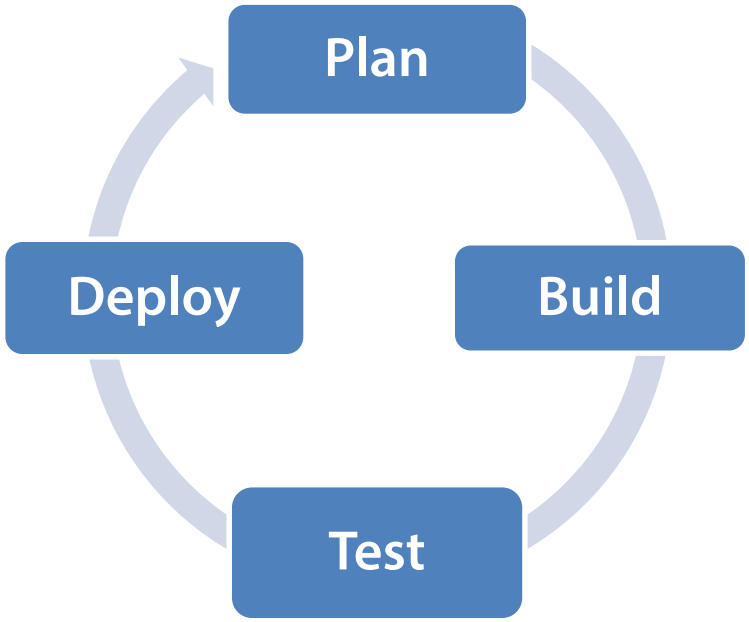
Design Meetings

Development Methods

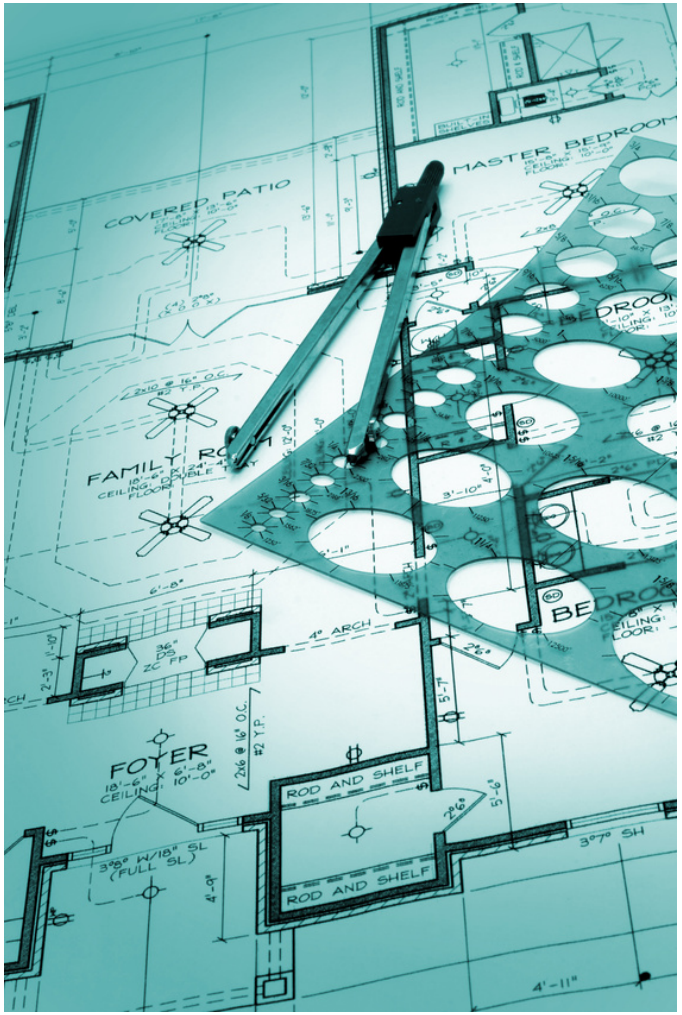
Waterfall



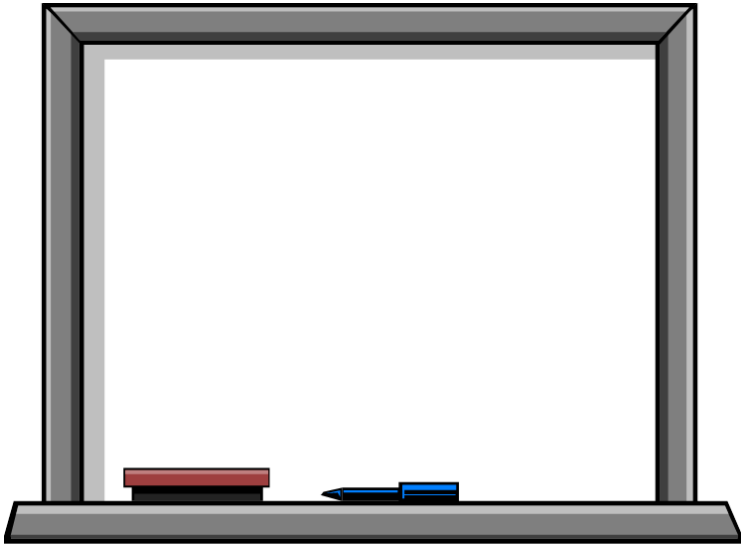
Iterative



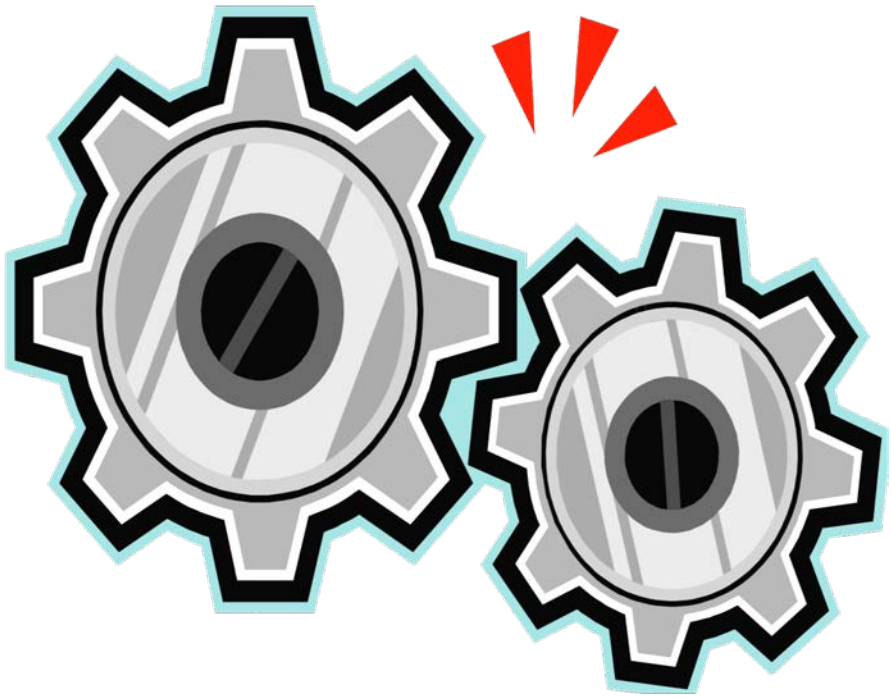
Formal Tools



Informal Tools

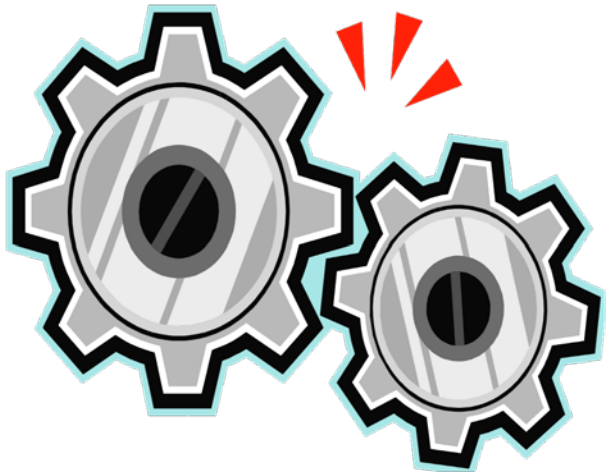


Behavioral Diagrams



Behavioral Diagrams

- Interactions
- Functionality
- Verbs and Actions

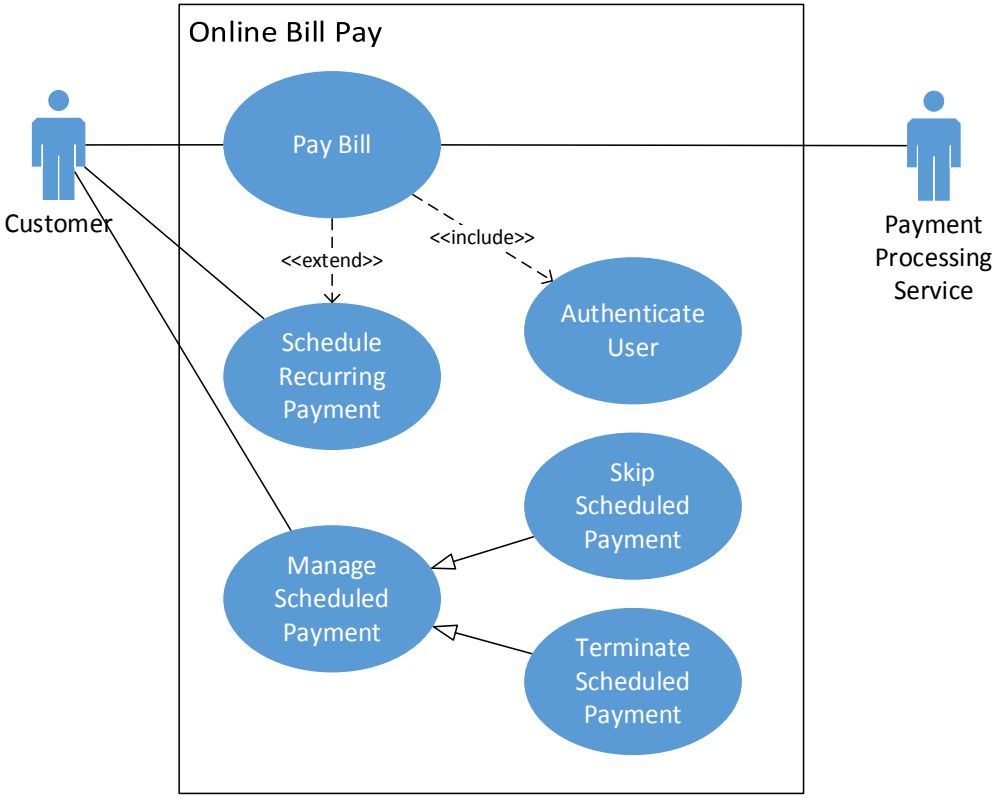


Introduction

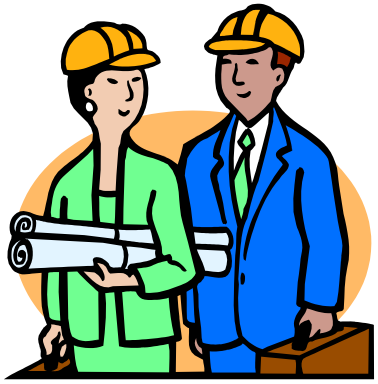
- Behavioral Diagrams
- Use Case Diagram
- Sequence Diagram
- State Diagram
- Activity Diagram

Use Case Diagram

- User Tasks
- System Interactions
- What not How

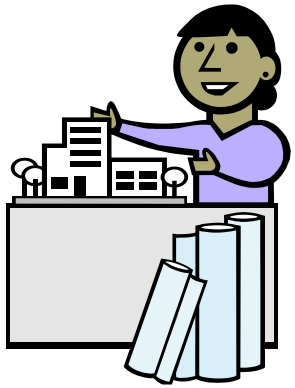


Common Users of Use Case Diagrams



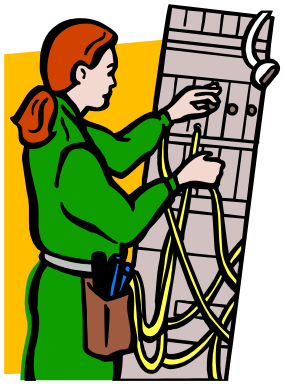
Product Owner

Business Analyst

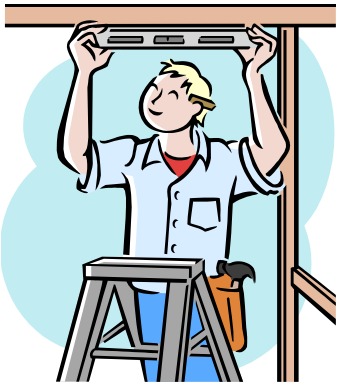


Architect

Operations



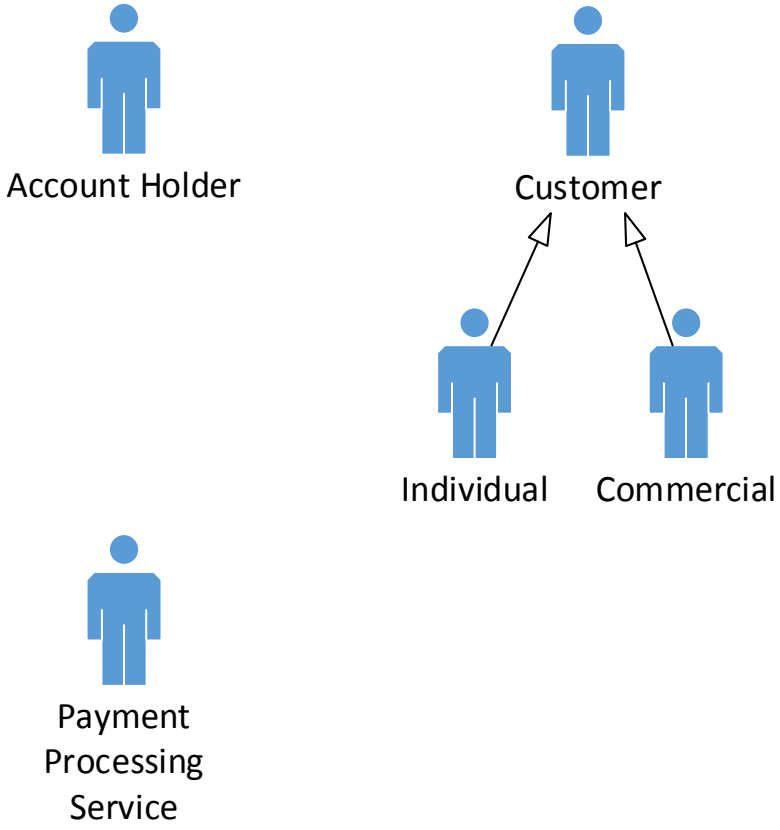
Quality Assurance



Developer

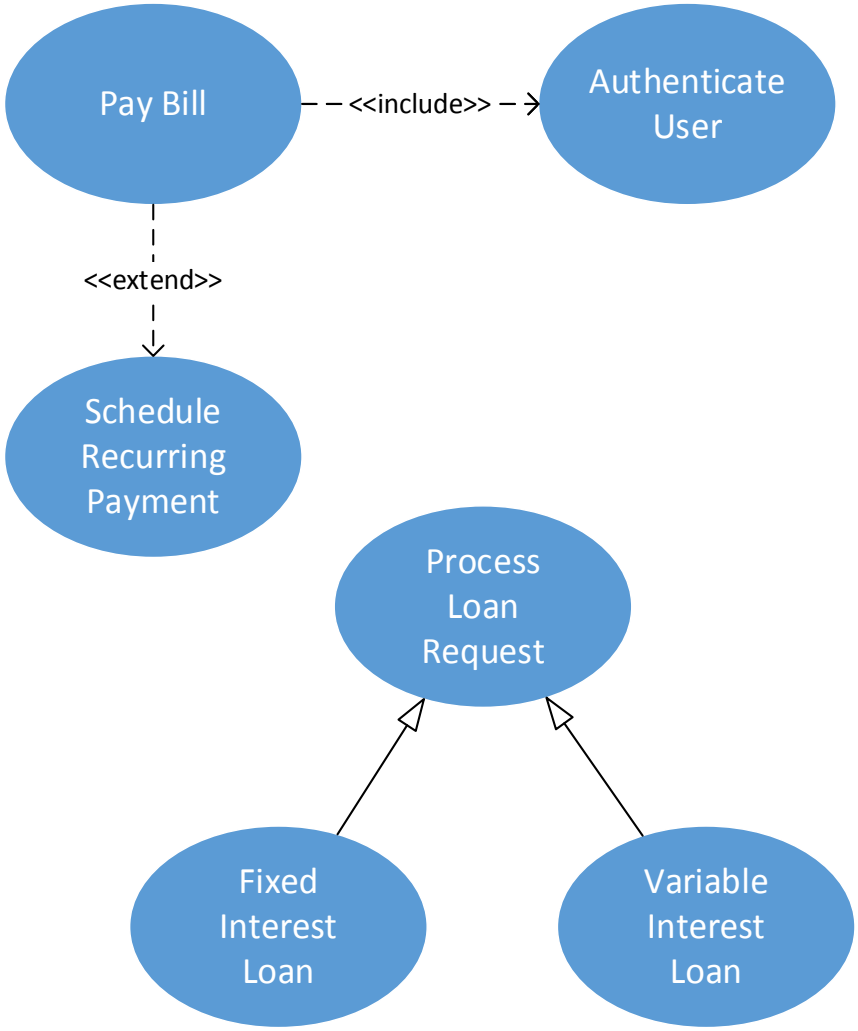
Actors

- **People**
  - Roles
- **Generalization**
- **Systems**



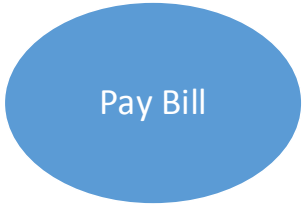
Use Cases

- Simple name
- User Tasks
- System Interactions
- Factor out common processes (<<include>>)
- Identify optional additions (<<extend>>
- Generalize

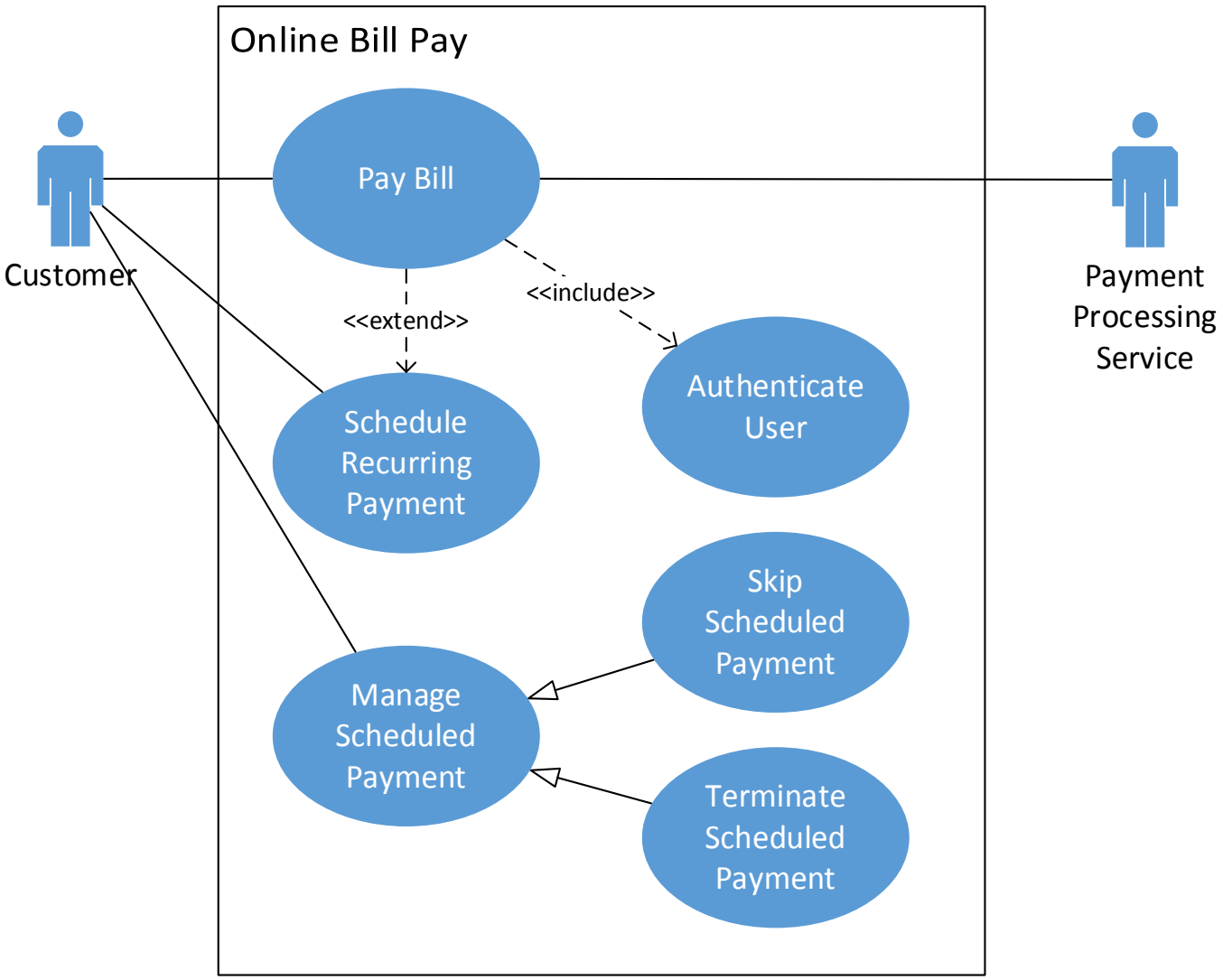


Scenarios

- **One Use Case may have many scenarios**
- **Scenario**
  - Steps in process
  - Branches
  - Extensions
  - Exceptions
- **Use Case vs. Use Case Diagram**
  - Pre-Conditions
  - Post-Conditions
- **May Relate to User Stories**

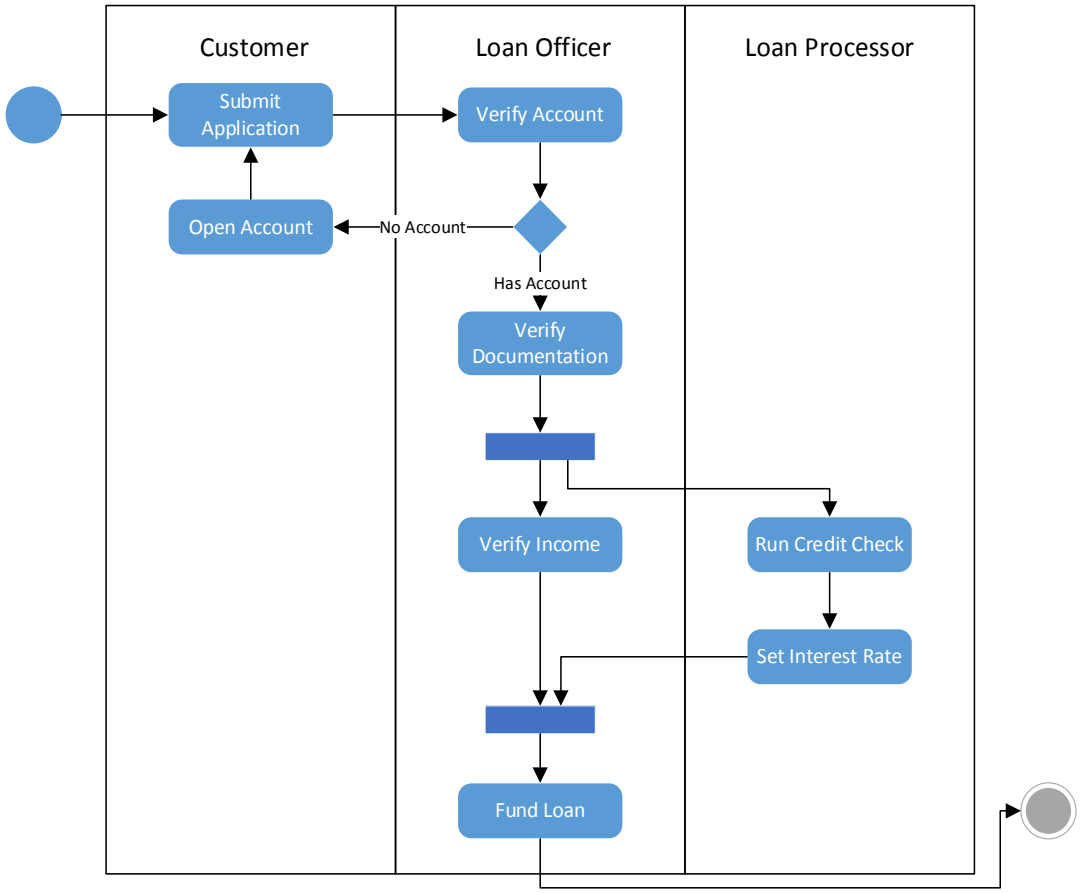


Basic Use Case Diagram

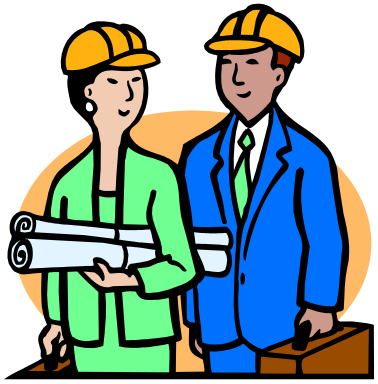


Activity Diagram

- Workflows
- Operations

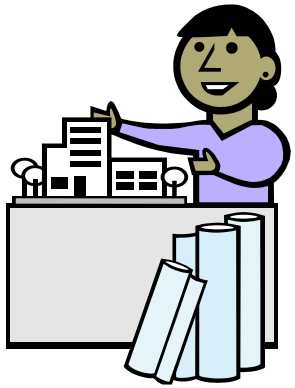


Common Users of Activity Diagrams



Product Owner

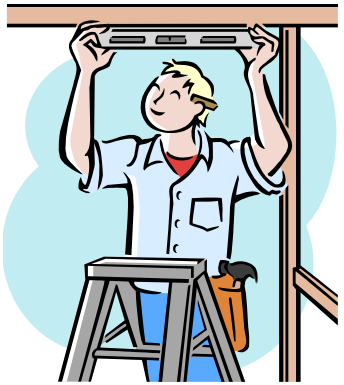
Business Analyst



Architect

Operations

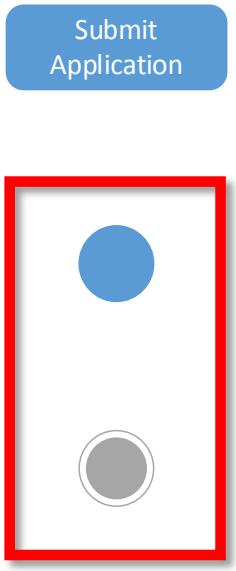
Quality Assurance



Developer

Actions and Activities

- **Actions**
  - Single Step
- **Activity**
  - Multiple Step
    - Decomposed in own diagram
- **Special**
  - Initiation
  - Completion

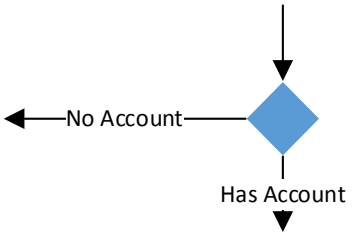


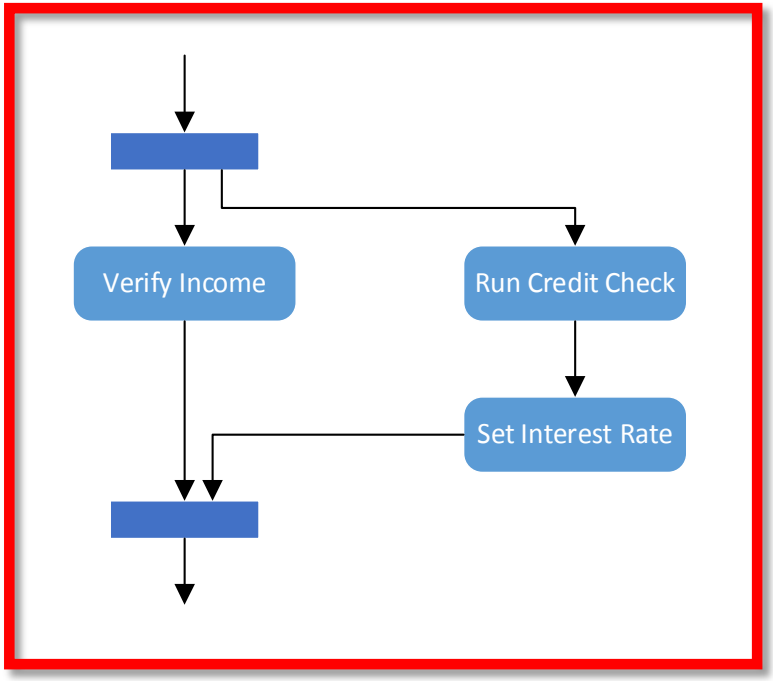
Flow Control

- **Decision / Branch**
- **Fork and Join**

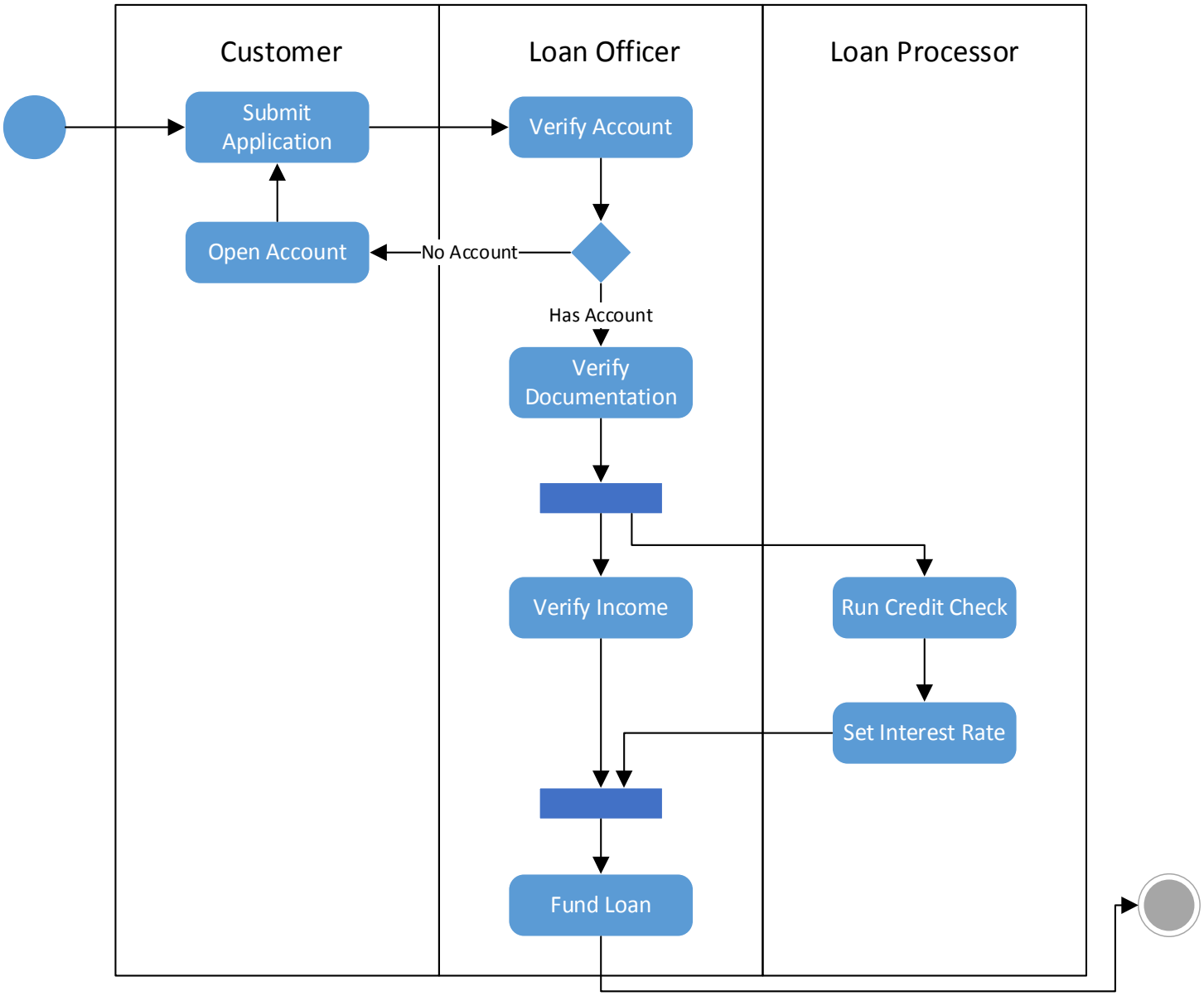
  Parallel / Concurrent

  Wait for all to join



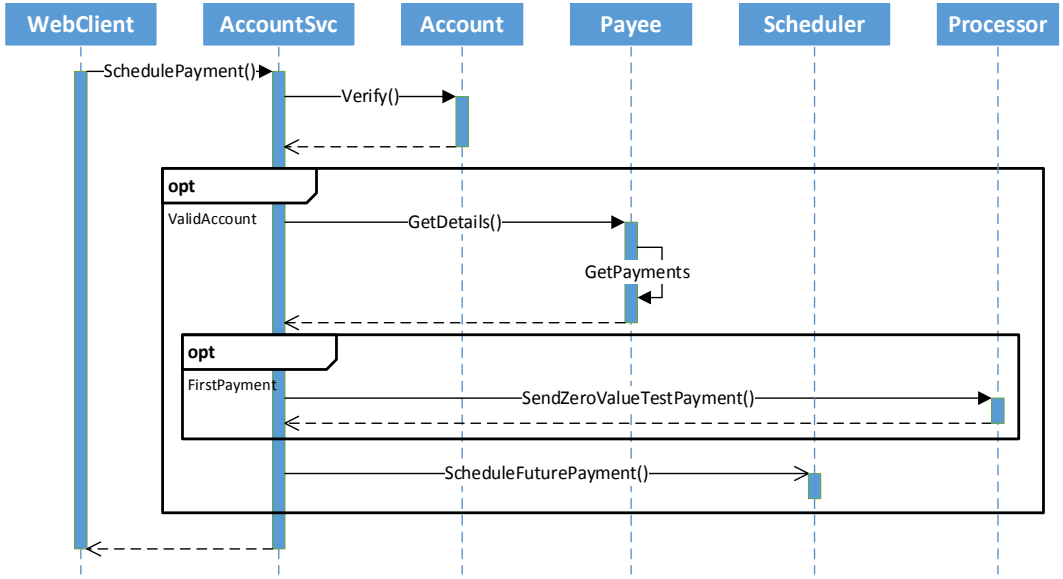


Basic Activity Diagram



Sequence Diagram

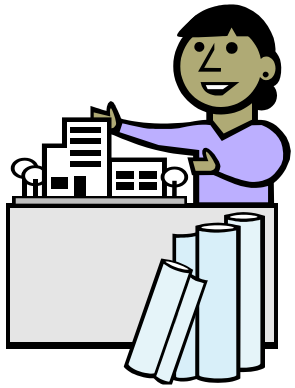
- Object Interaction
- Focus on time/order



Common Users of Sequence Diagrams

Product Owner

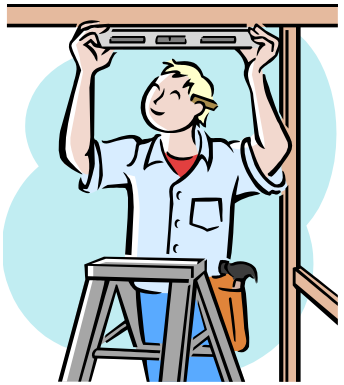
Business Analyst



Architect

Operations

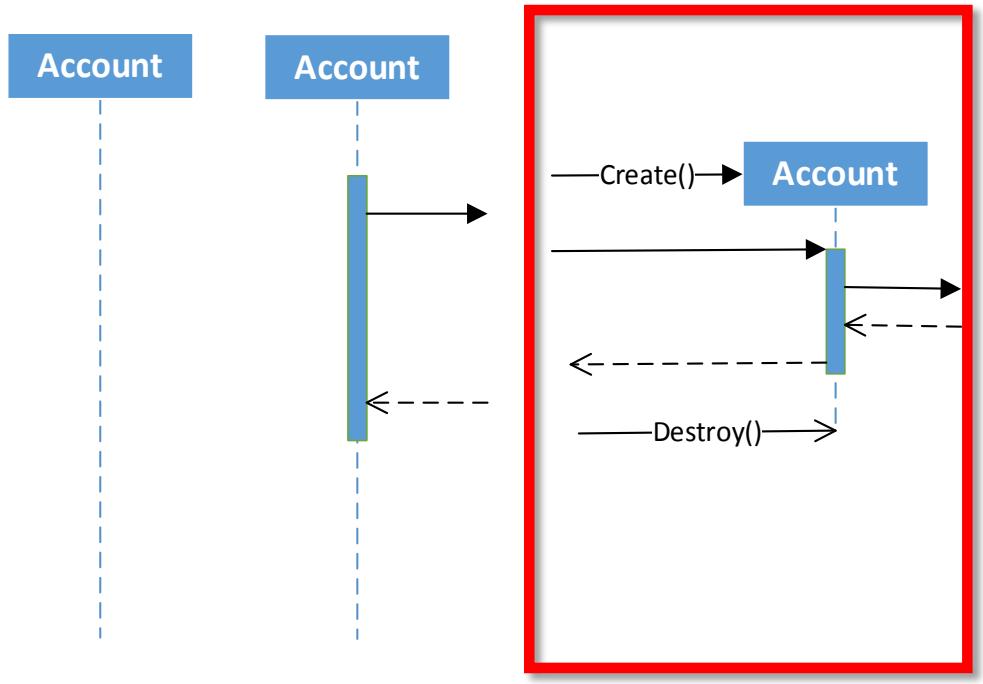
Quality Assurance



Developer

Classes

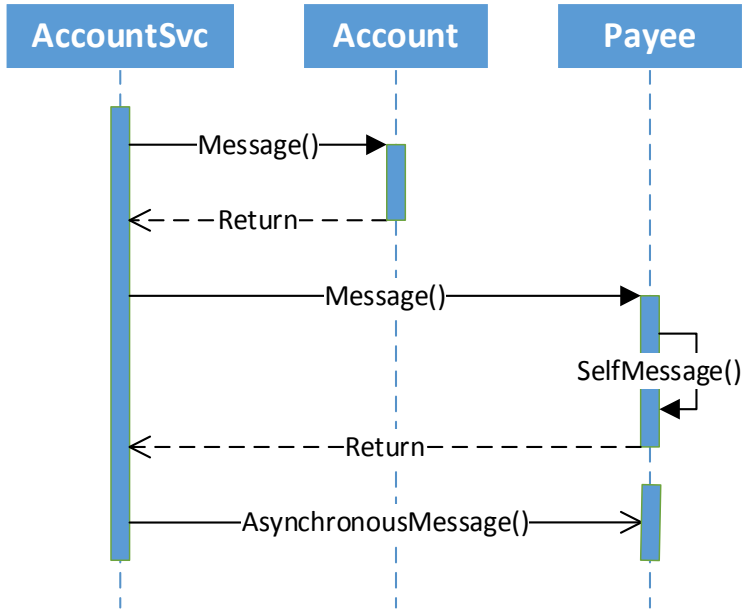
- **Lifeline**
- **Focus of Control**
- **Object Lifetime**
  - Creation
  - Termination



Messages

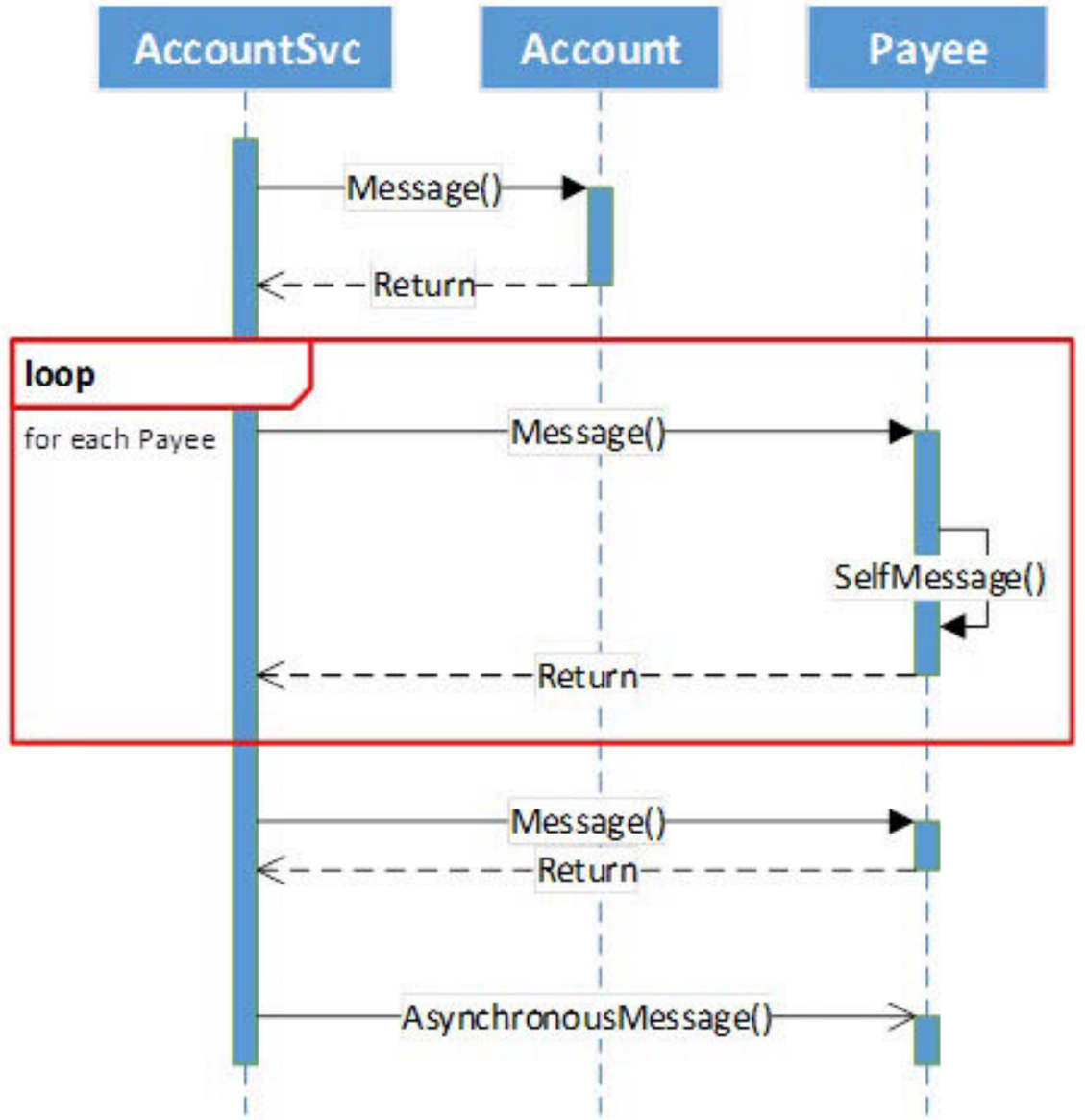
- **Message**
- **Return**
- **Self Message**
- **Asynchronous**

  No waiting for Return



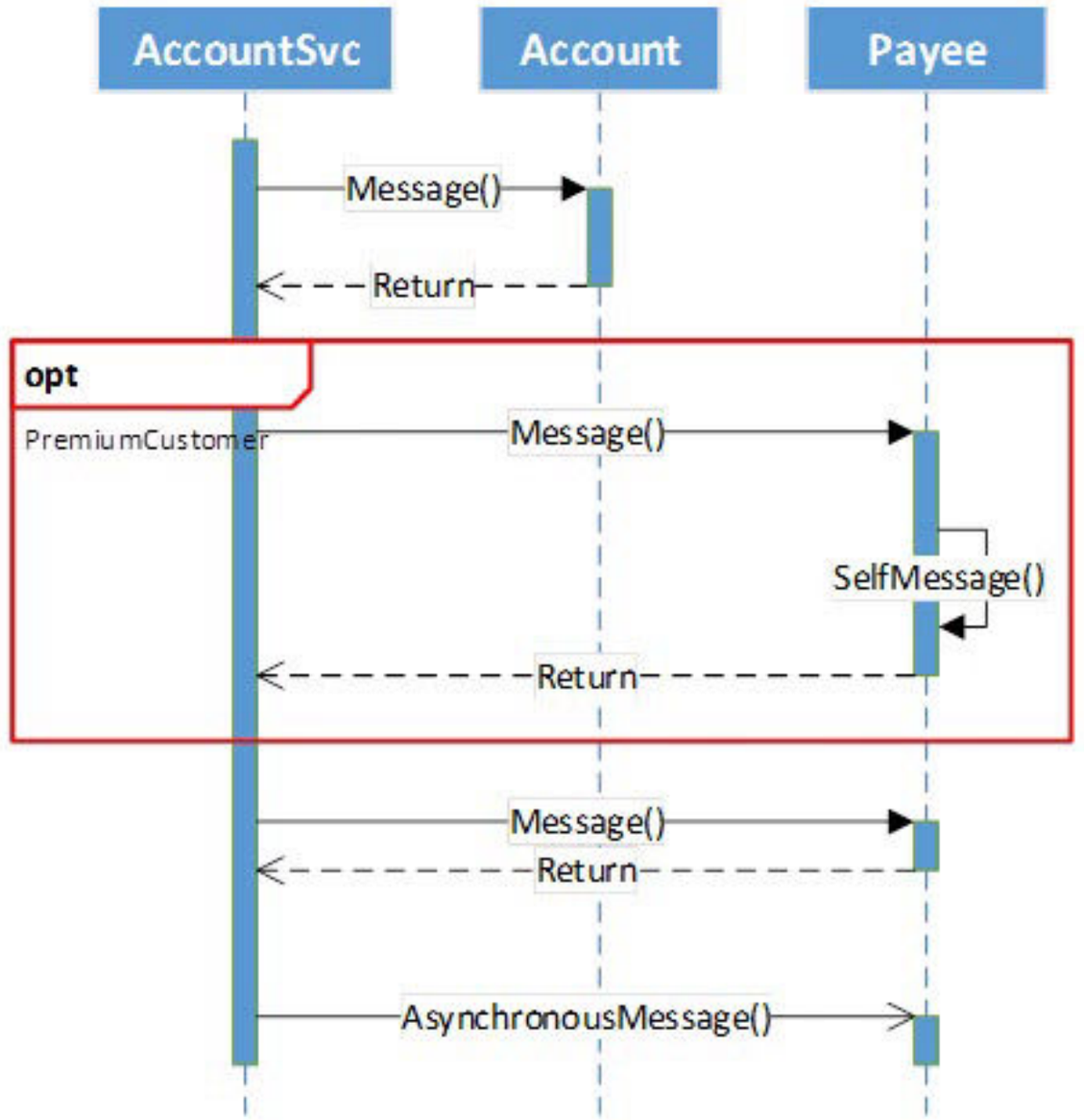
Structured Control

- Looping



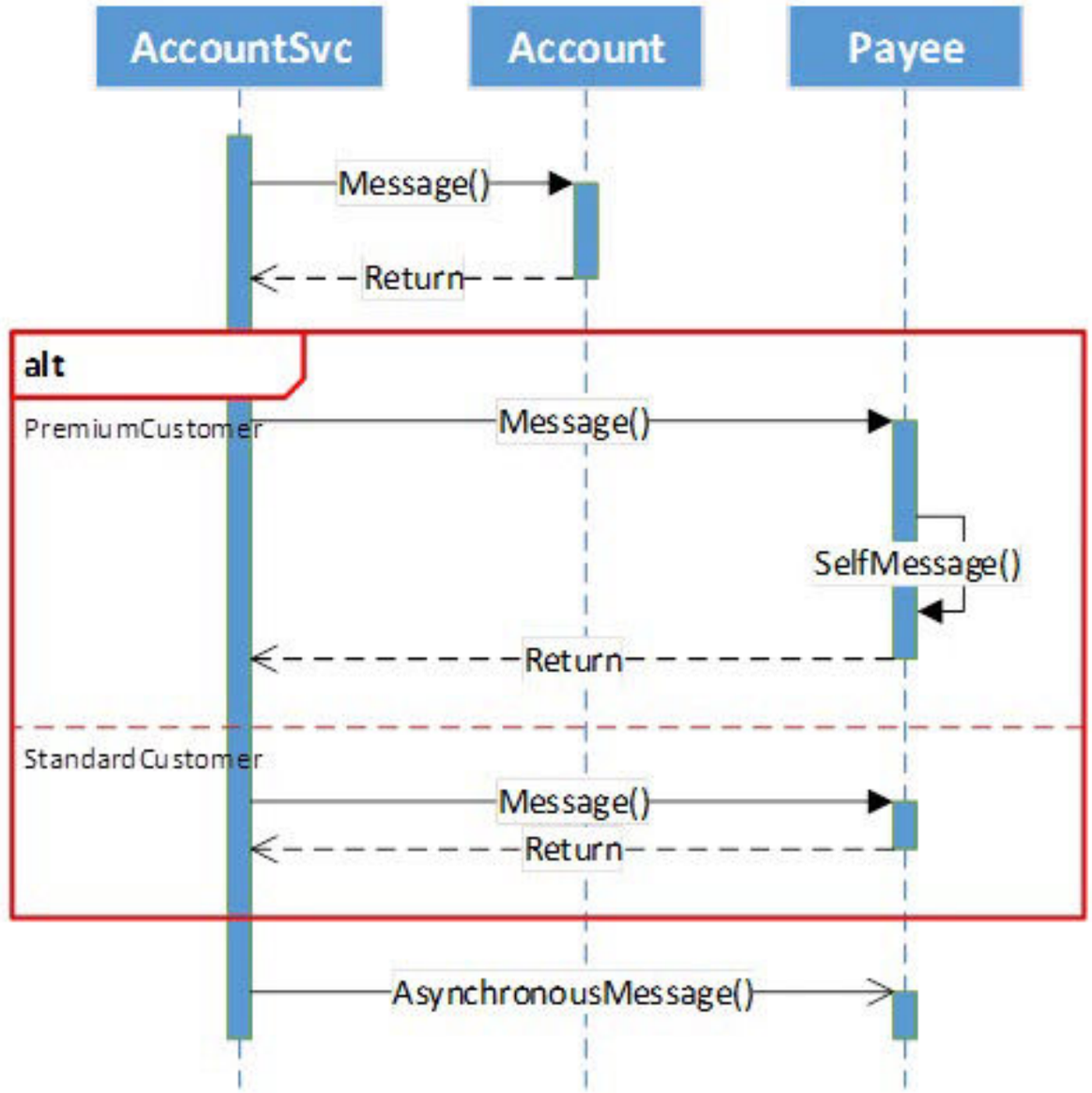
Structured Control

- Looping
- Optional



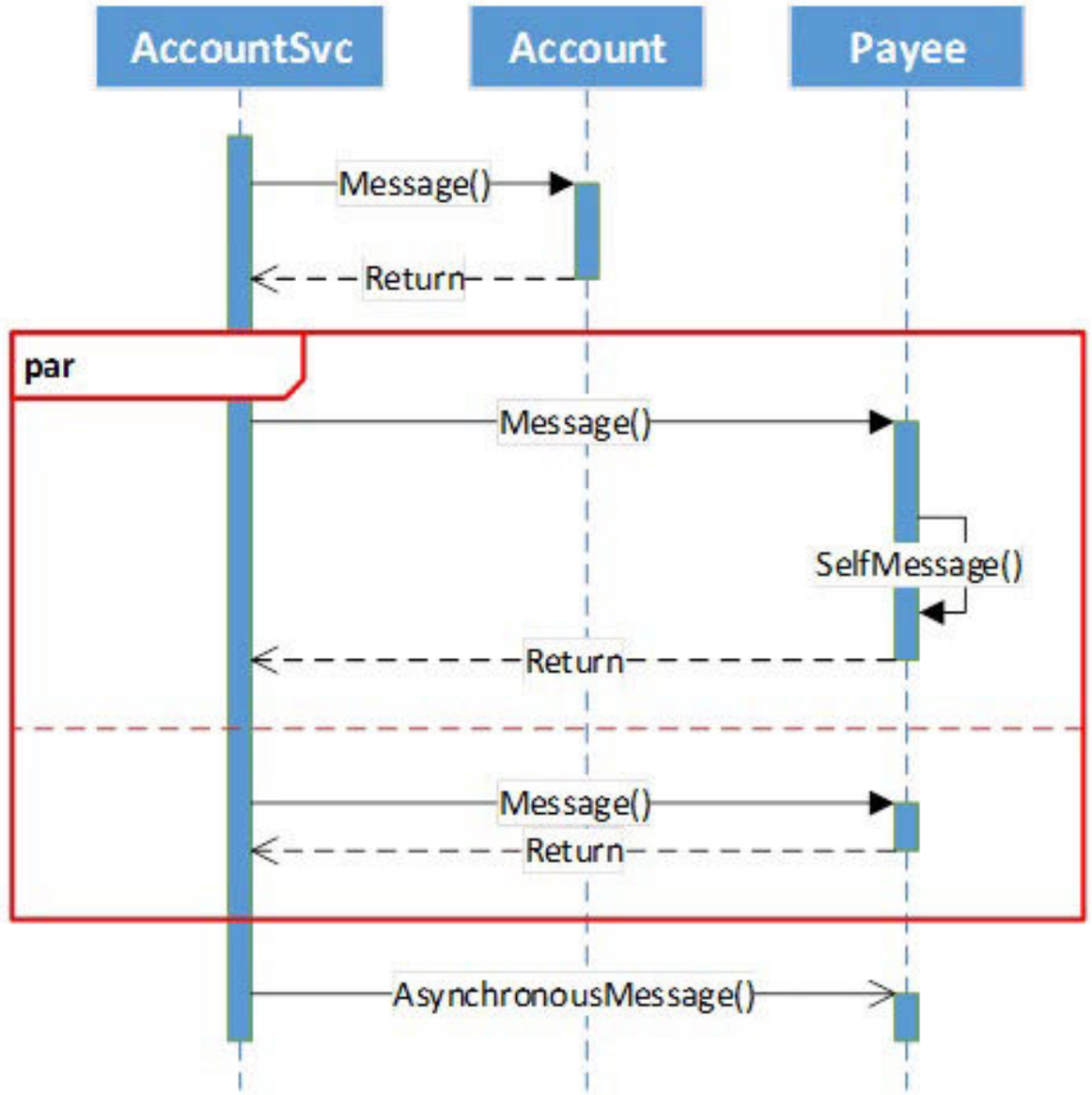
Structured Control

- Looping
- Optional
- Conditional
  - [else]

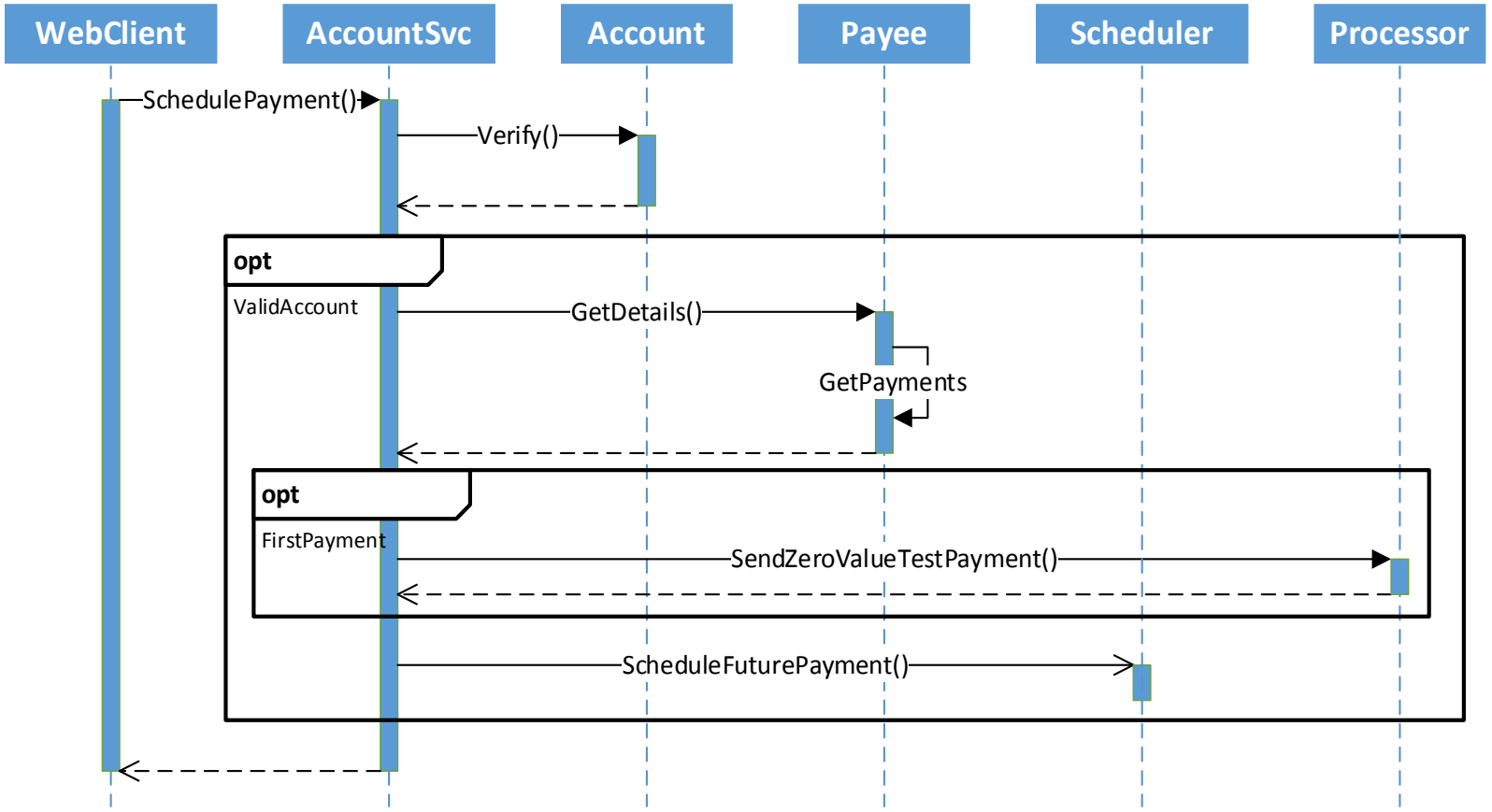


Structured Control

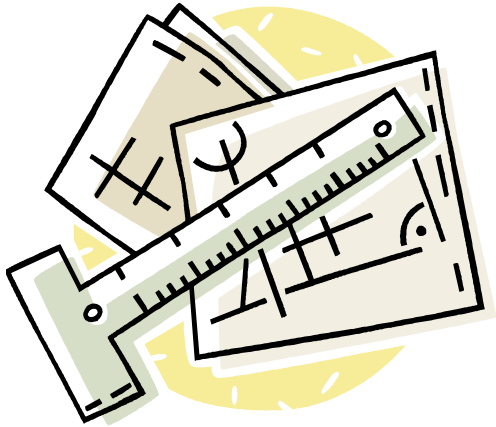
- Looping
- Optional
- Conditional
  - [else]
- Parallel



Basic Sequence Diagram



Structural Diagrams



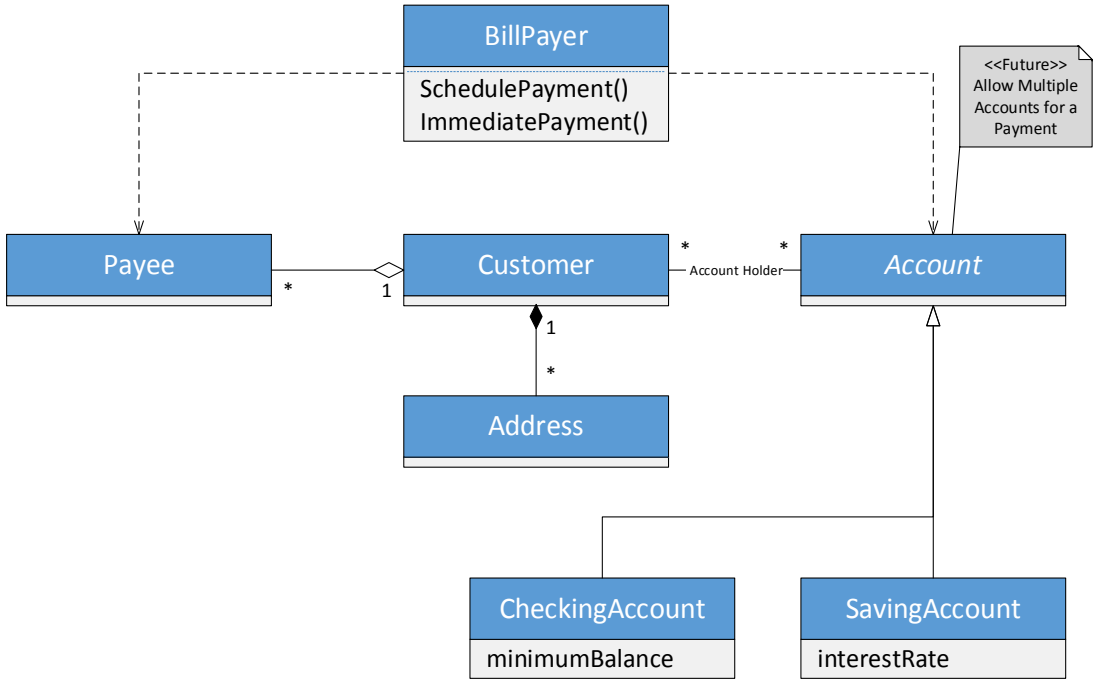
Structural Diagrams

- Project Vocabulary
- Structural Relationships
- Architectural Diagrams

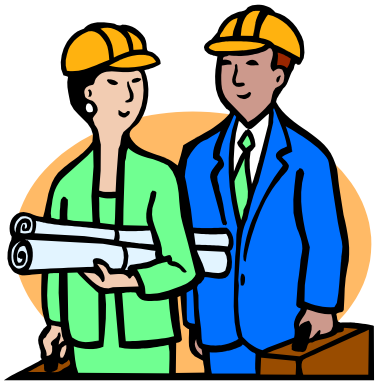
Class Diagram

- Vocabulary
- Relationships

"Boxes and Lines"

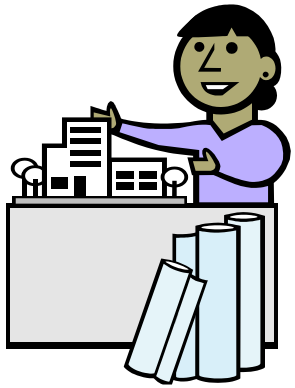


Common Users of Class Diagrams



Product Owner

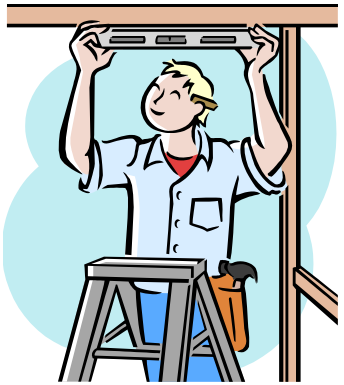
Business Analyst



Architect

Operations

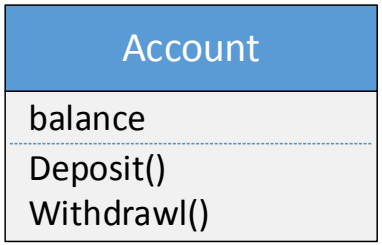
Quality Assurance

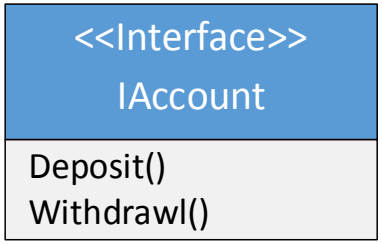


Developer

Classes and Interfaces

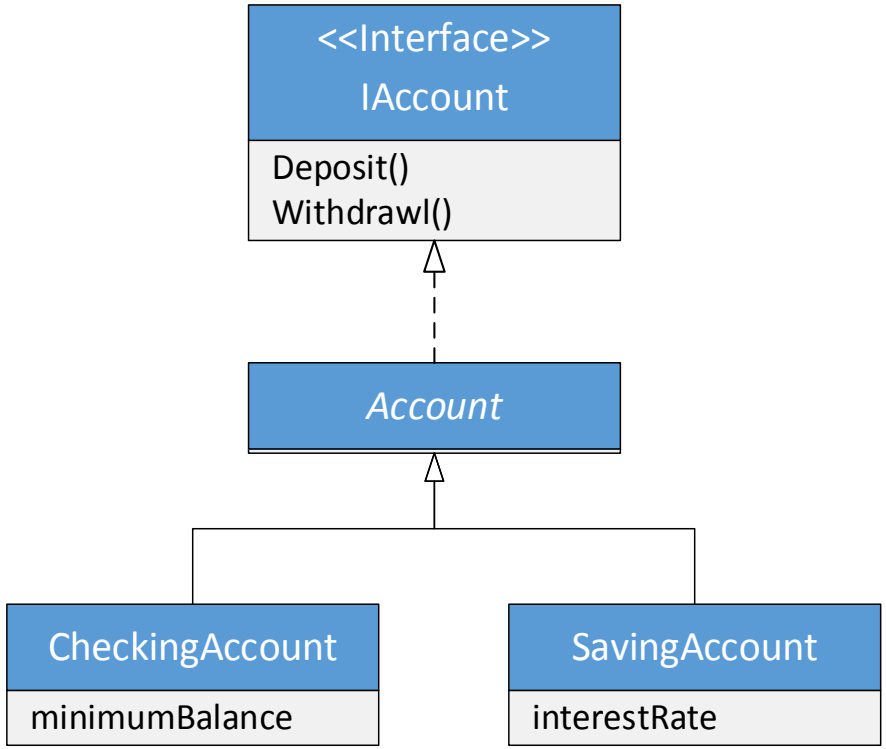
- **Class**
  - Represents an Entity (Noun)
    - Attributes
    - Operations
  - Omit Details if not required
- **Interface**





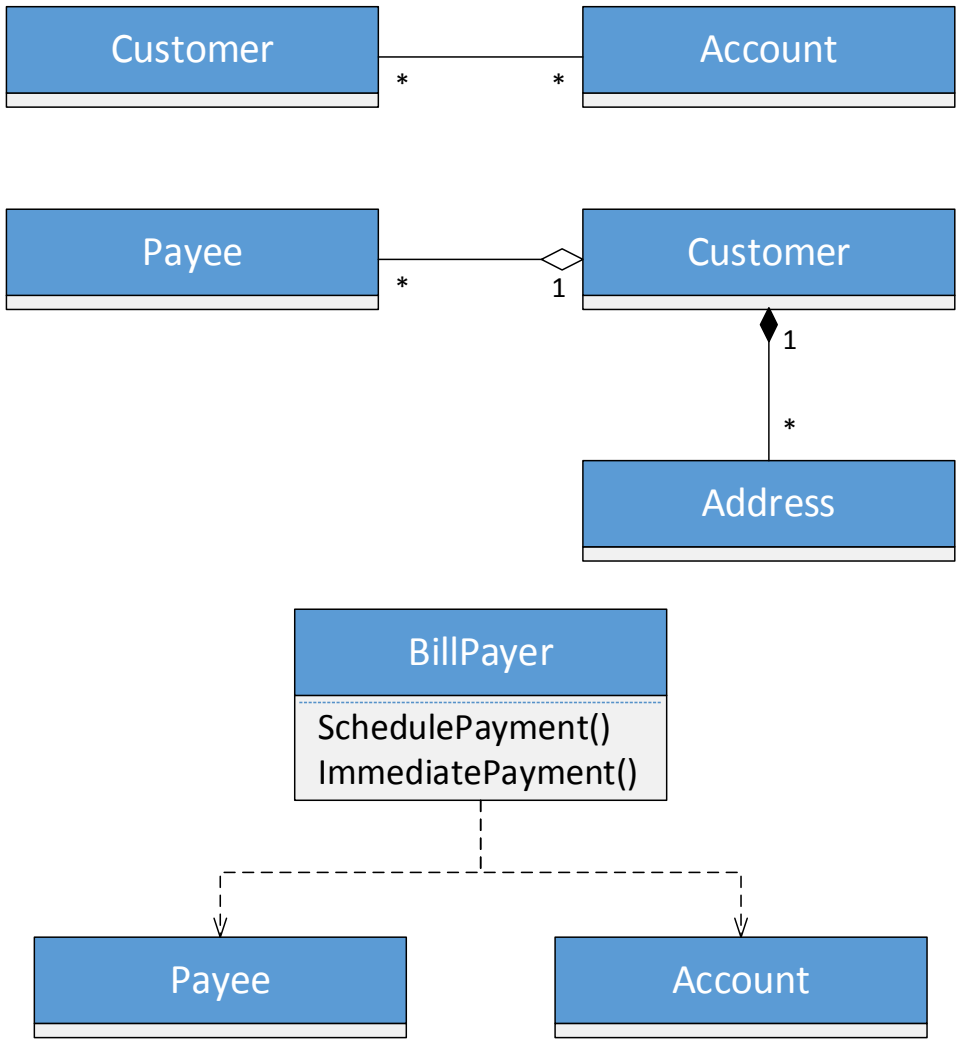
Implementation and Inheritance

- **Implement Interface**
- **Inheritance ("Is A")**
  - Abstract Class
  - Concrete Class

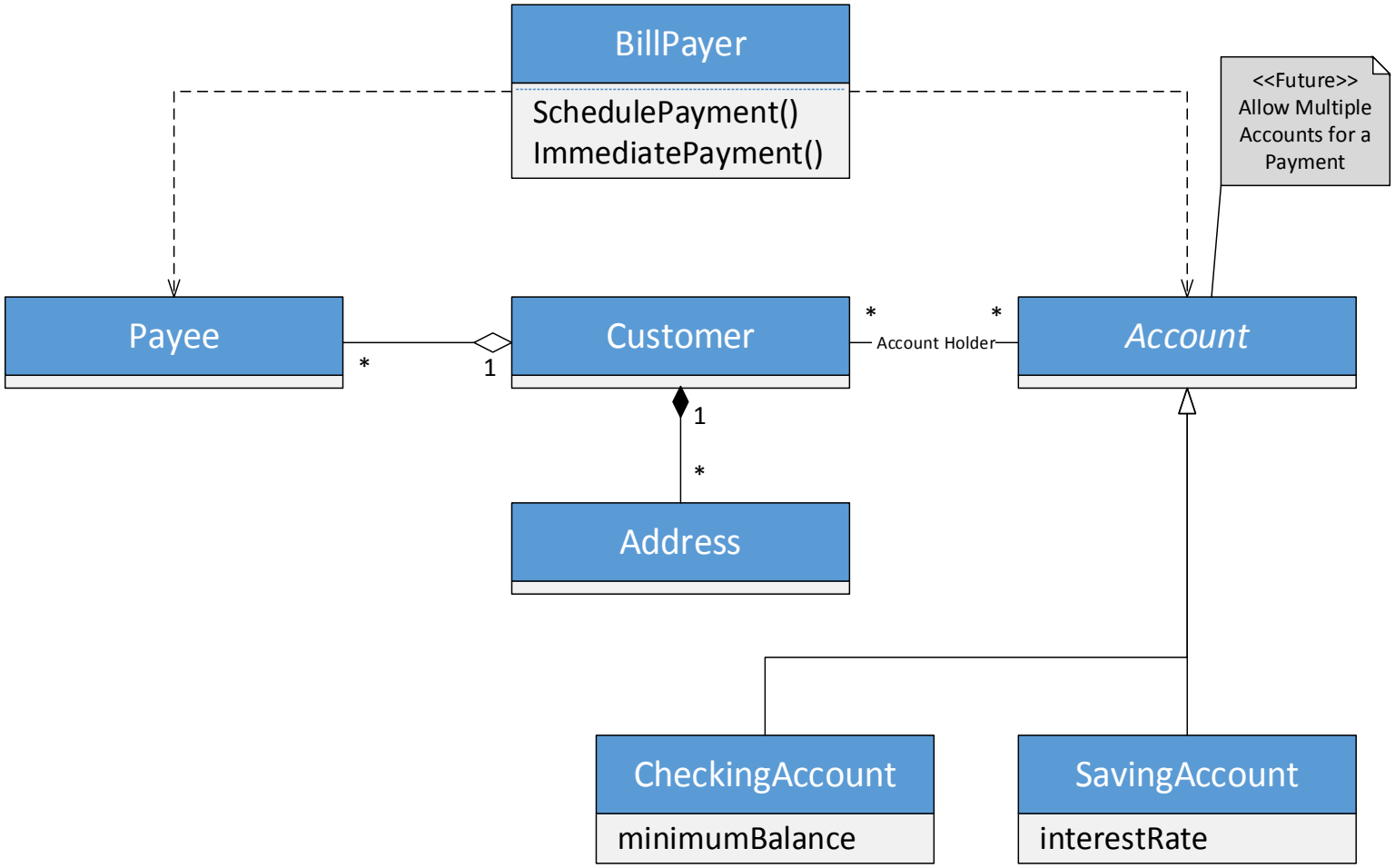


Relationships

- **Basic ("Has A")**
  - Multiplicity
- **Aggregation**
  - Whole - Part
- **Composition**
  - Ownership
- **Uses**

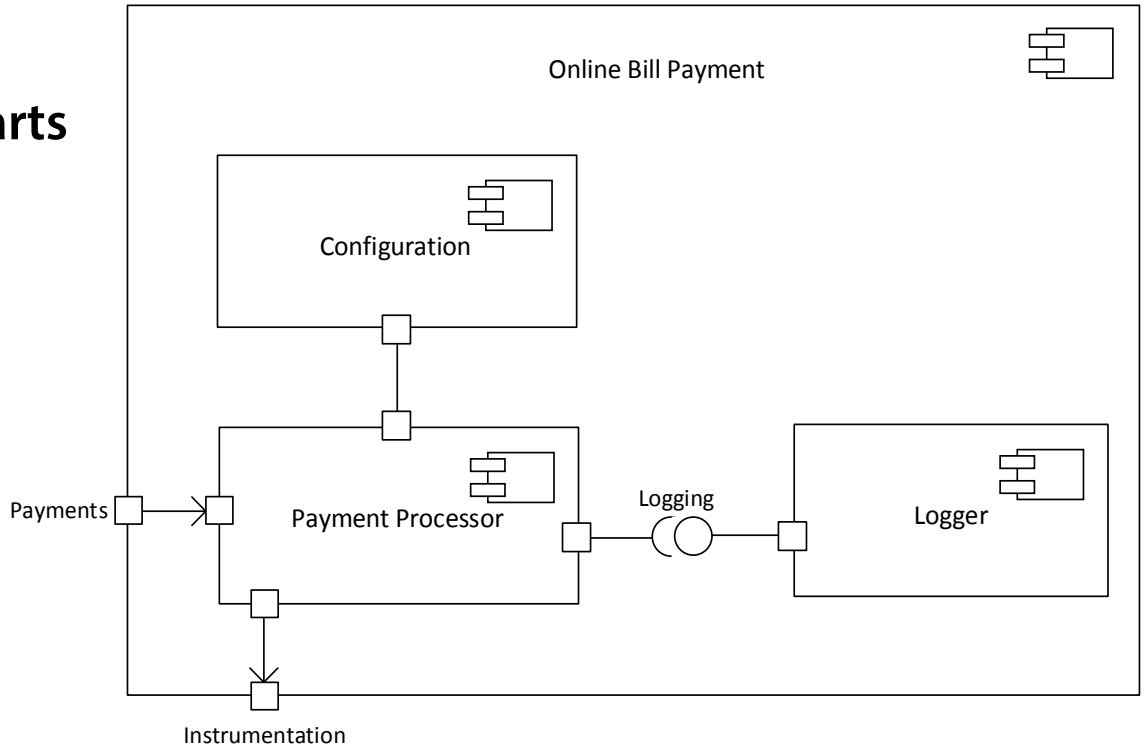


Basic Class Diagram



Component Diagram

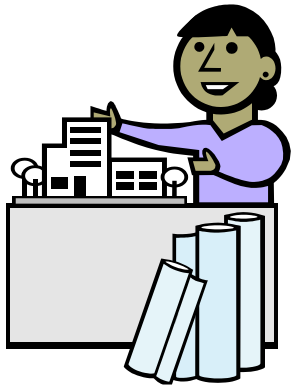
- Identify Interfaces
- Define replaceable parts



Common Users of Component Diagrams

Product Owner

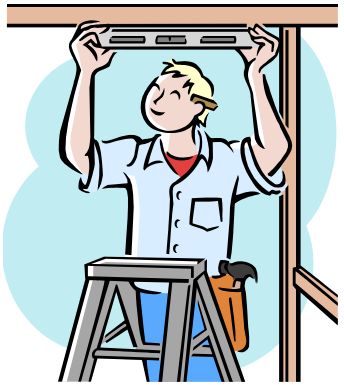
Business Analyst



Architect

Operations

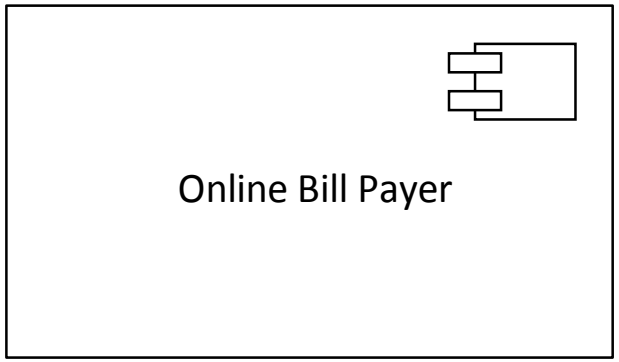
Quality Assurance

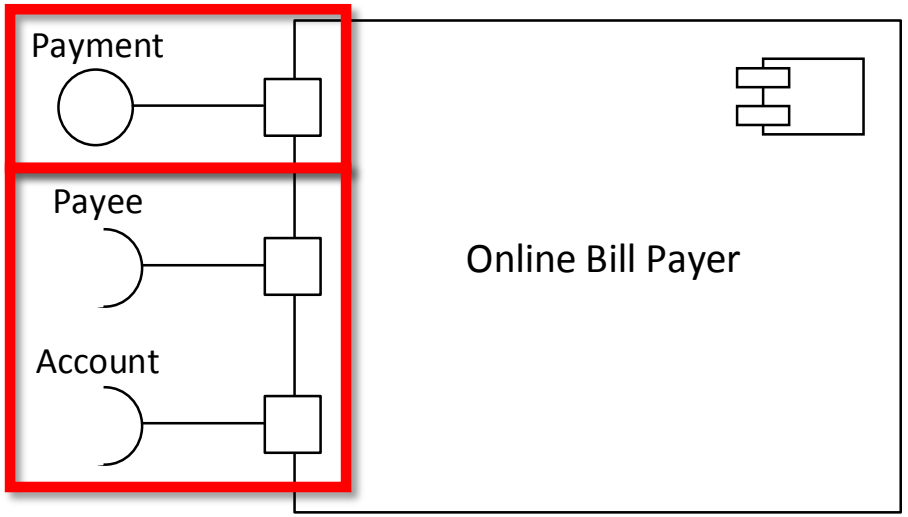


Developer

Basic Component

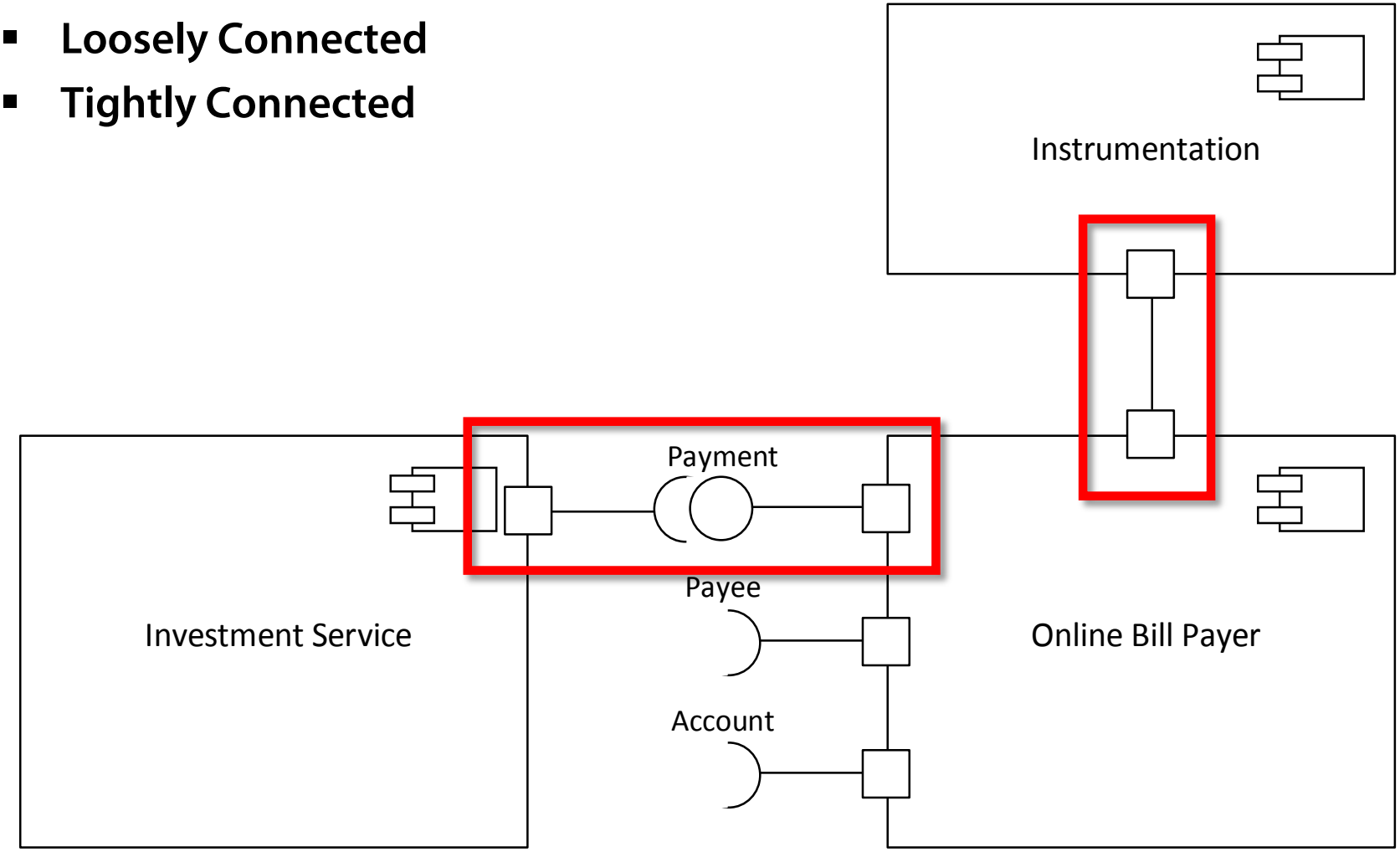
- **Component**
- **Interfaces**
  - Realized
  - Required



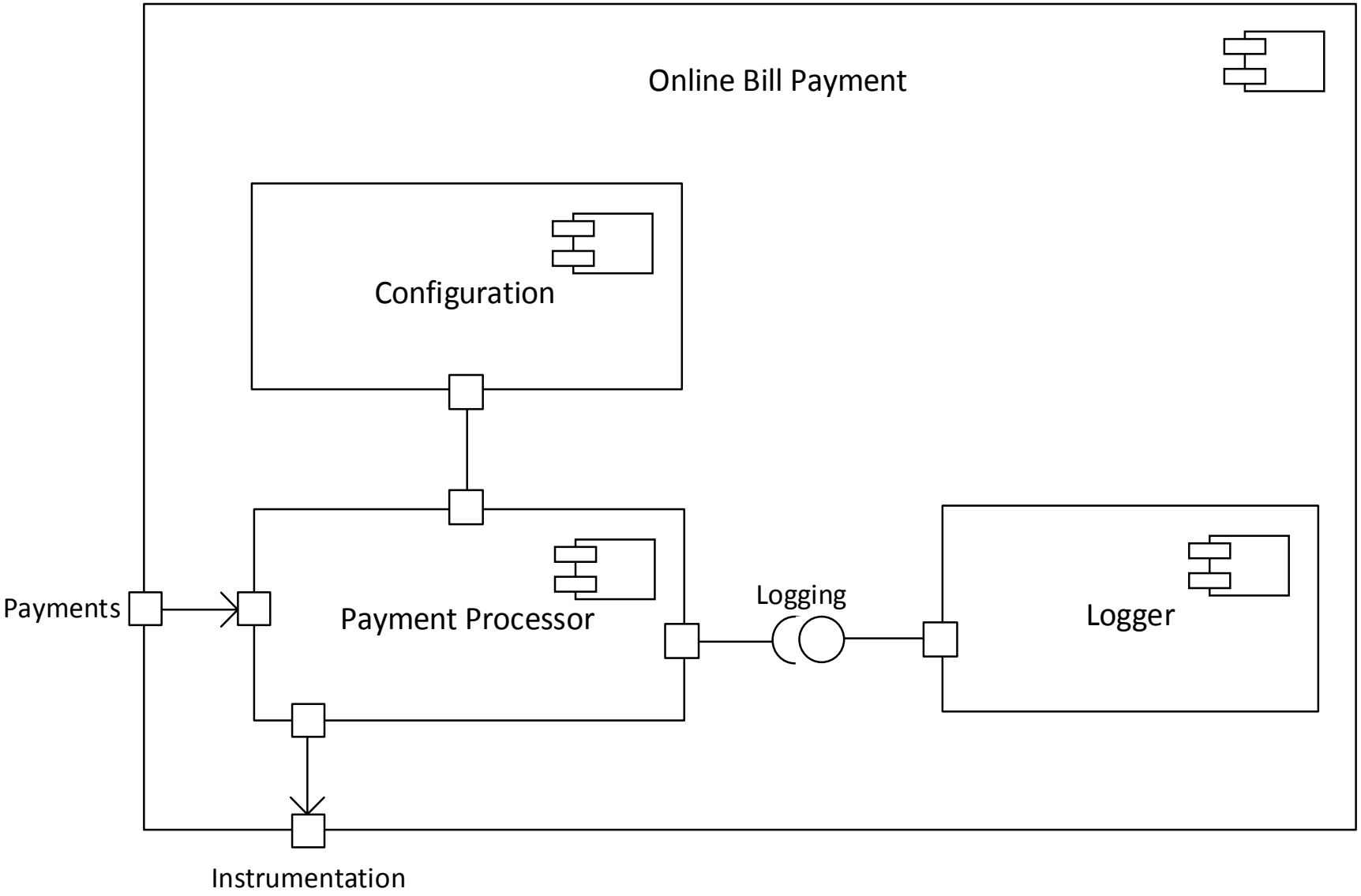


Connected Components

- Loosely Connected
- Tightly Connected

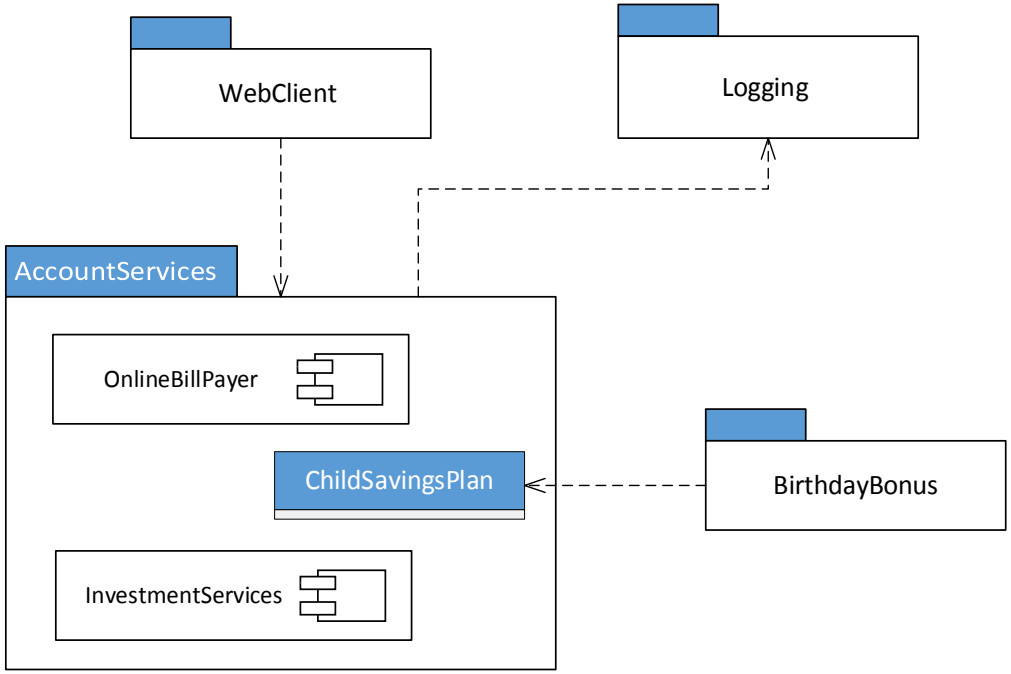


Basic Component Diagram



Package Diagram

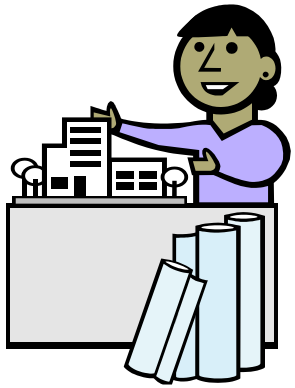
- Logical Container
- Support Large Models



Common Users of Package Diagrams

Product Owner

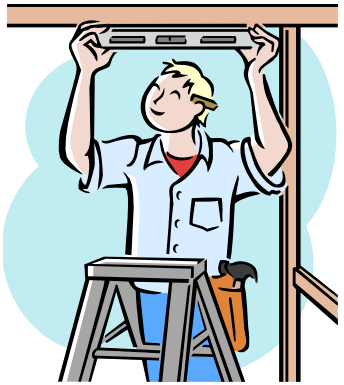
Business Analyst



Architect

Operations

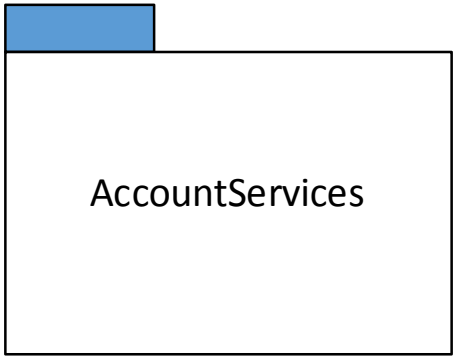
Quality Assurance

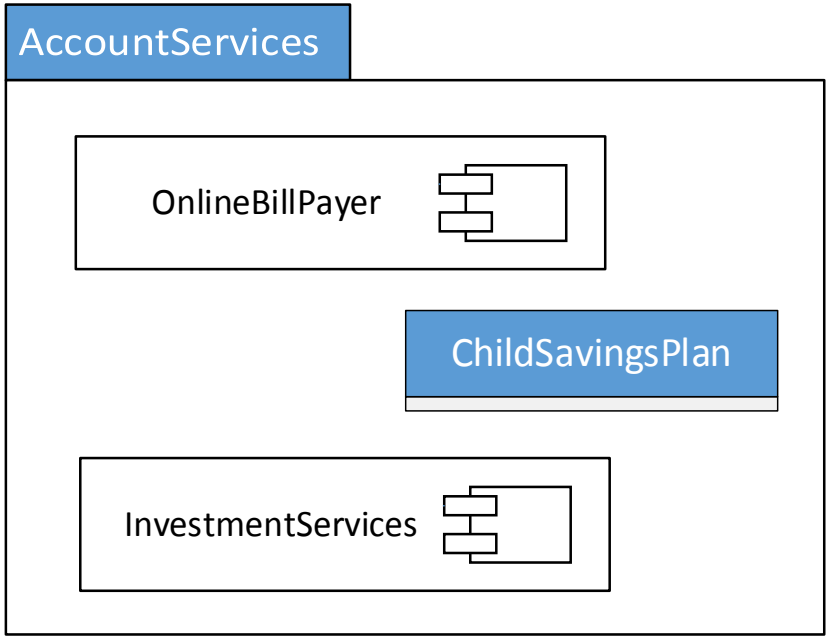


Developer

Simple Packages

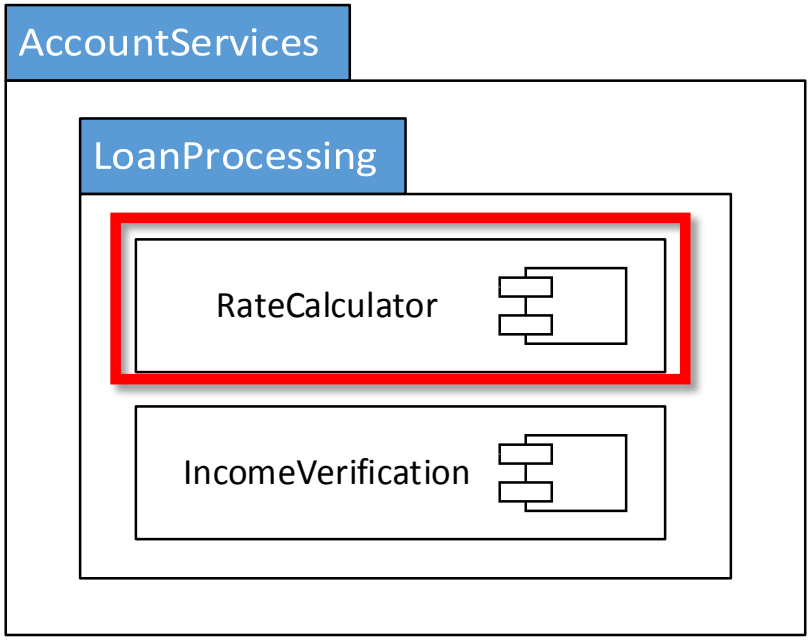
- **Basic**
- **Showing (relevant) elements**
- **Can Contain**
  - Classes
  - Interfaces
  - Components
  - Nodes
  - Diagrams
  - Packages
  - Etc.





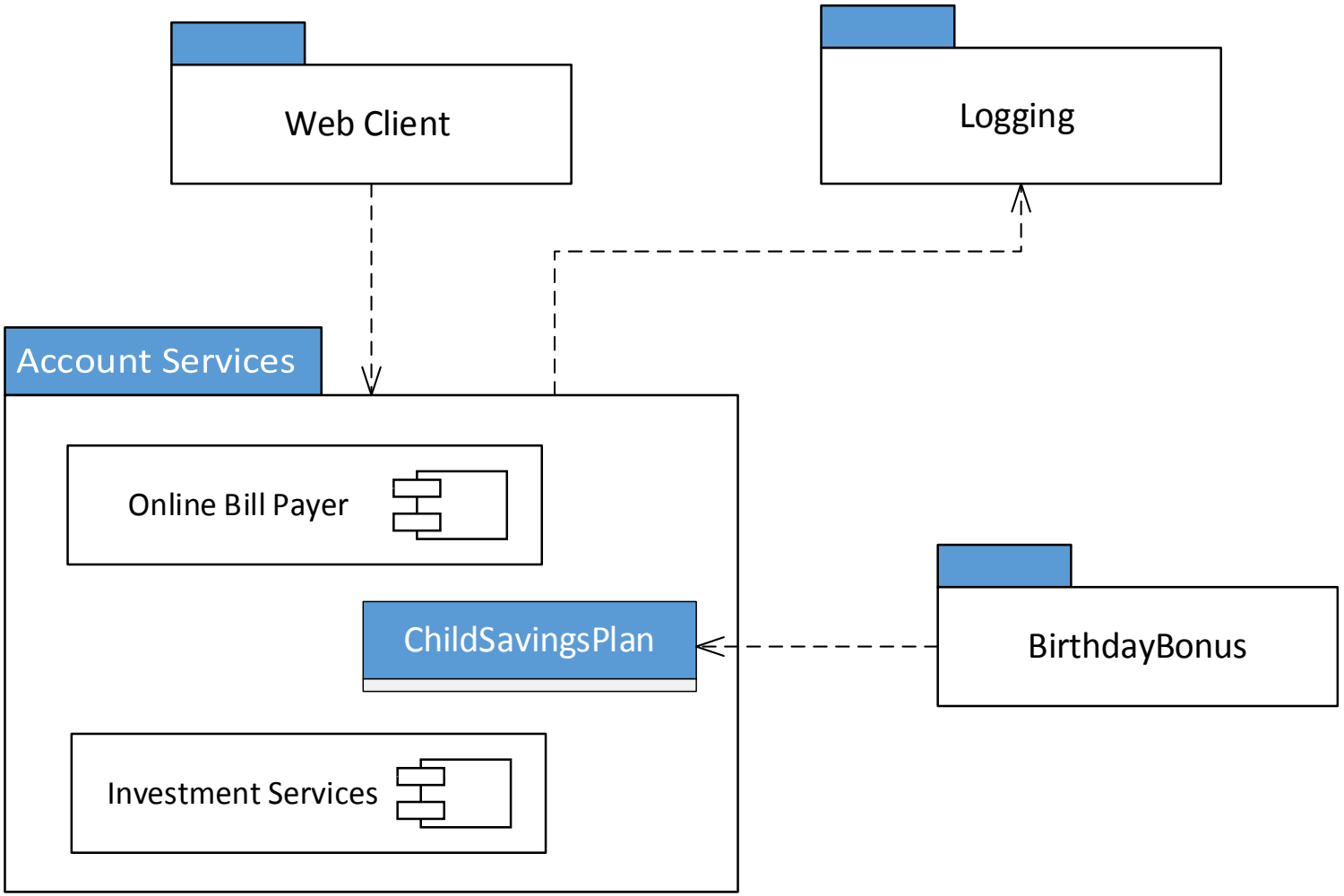
Nested Packages

- **Contained Packages**
  - Maximum 2 or 3 deep
- **Packages define scope**
- **Qualified Names**
  - Use double colon - ::



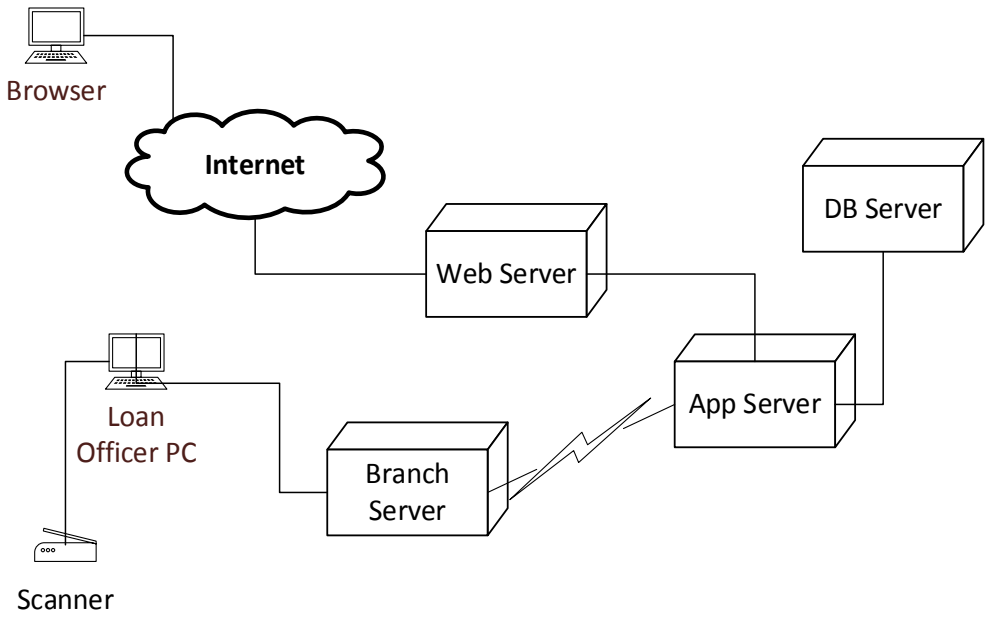
AccountServices::LoanProcessing::RateCalculator

Basic Package Diagram

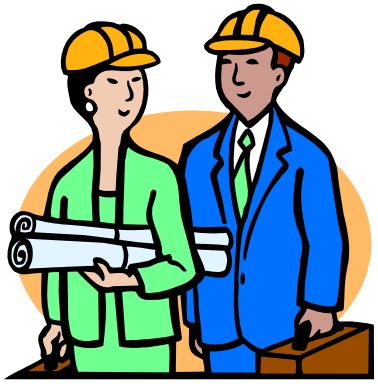


Deployment Diagram

- Map Components to Physical Systems
- View Special Devices
- Visualize Hardware and Network Dependencies

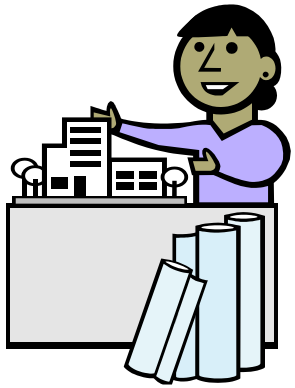


Common Users of Deployment Diagrams



Product Owner

Business Analyst



Architect

Operations

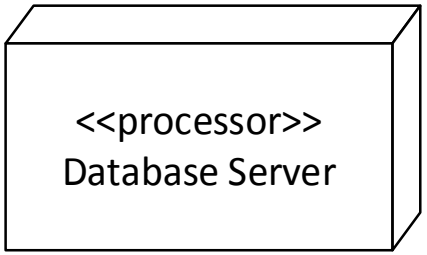
Quality Assurance

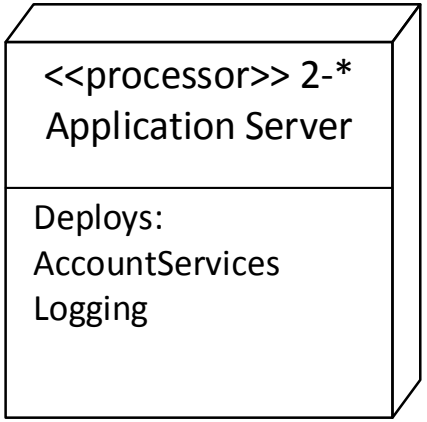
Developer

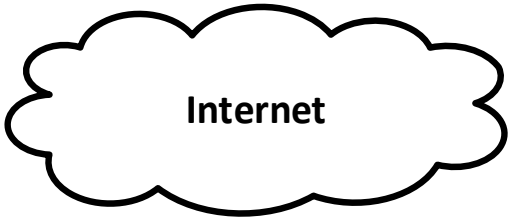
Nodes

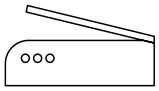
- **Processors**
  - Cubes
  - Stereotyped
  - Detailed
- **Devices**
  - Networks
  - Specialty

Web Server









Scanner

Basic Deployment Diagram