Functional Viewpoints

Lotfi ben Othmane

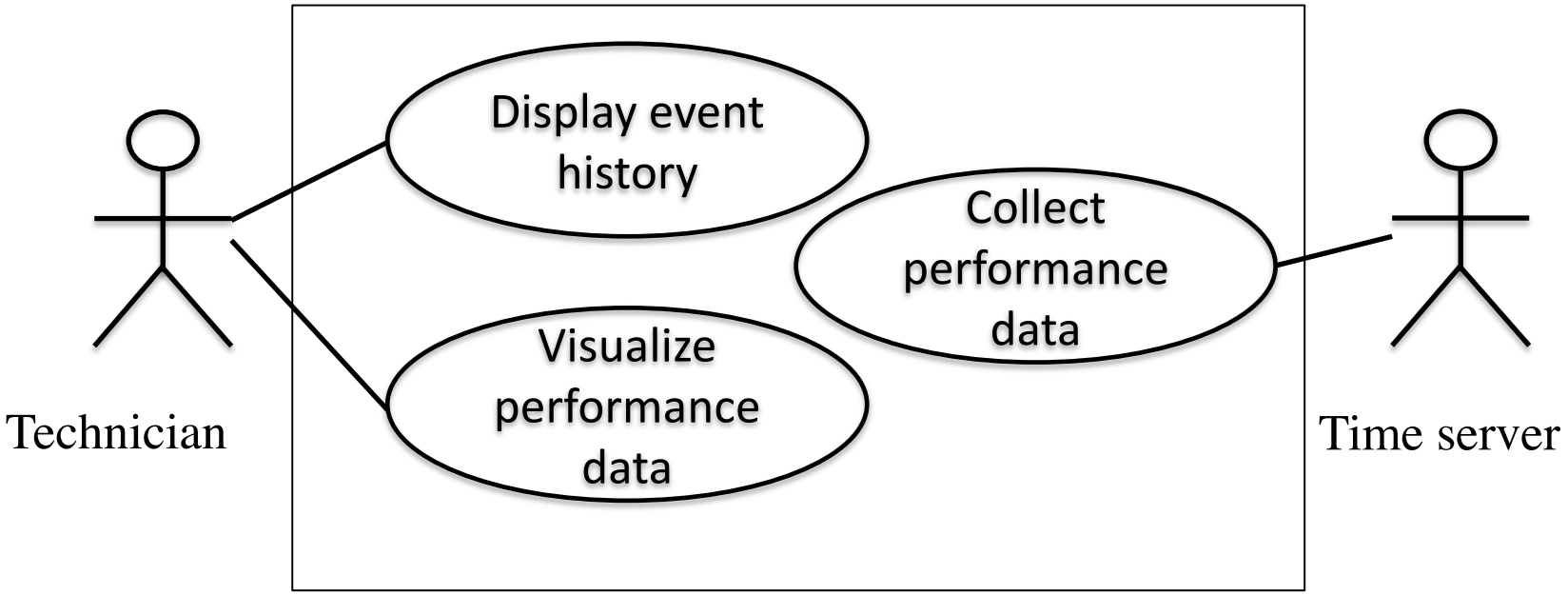
Example – Home Security System

Develop an architecture for home security system. The main functionalities are:

1. Set a system on and off
2. Collect data using sensors and camera
3. Analyze the data to detect intruders
4. Make alerts
5. Send alerts to central system
6. Keep data to assess dangerous neighborhood
7. Store structured data into an Oracle database
8. Store images in a MongoDB database

Primary Functionality

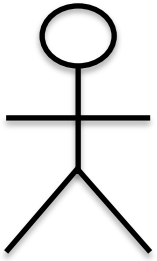
We use use cases to represent the functionalities of a system



Use Case Diagram

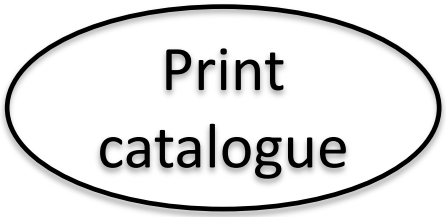
- System boundary: Rectangle shape representing the boundary of the system.
- Actors: A role that a user plays with respect to the system

An actor could be a system



Actor

- Use case: A set of scenarios that describe the interactions of the actors with the system



Use case

Use Case Diagram

- Association: communication between an actor and a use case.

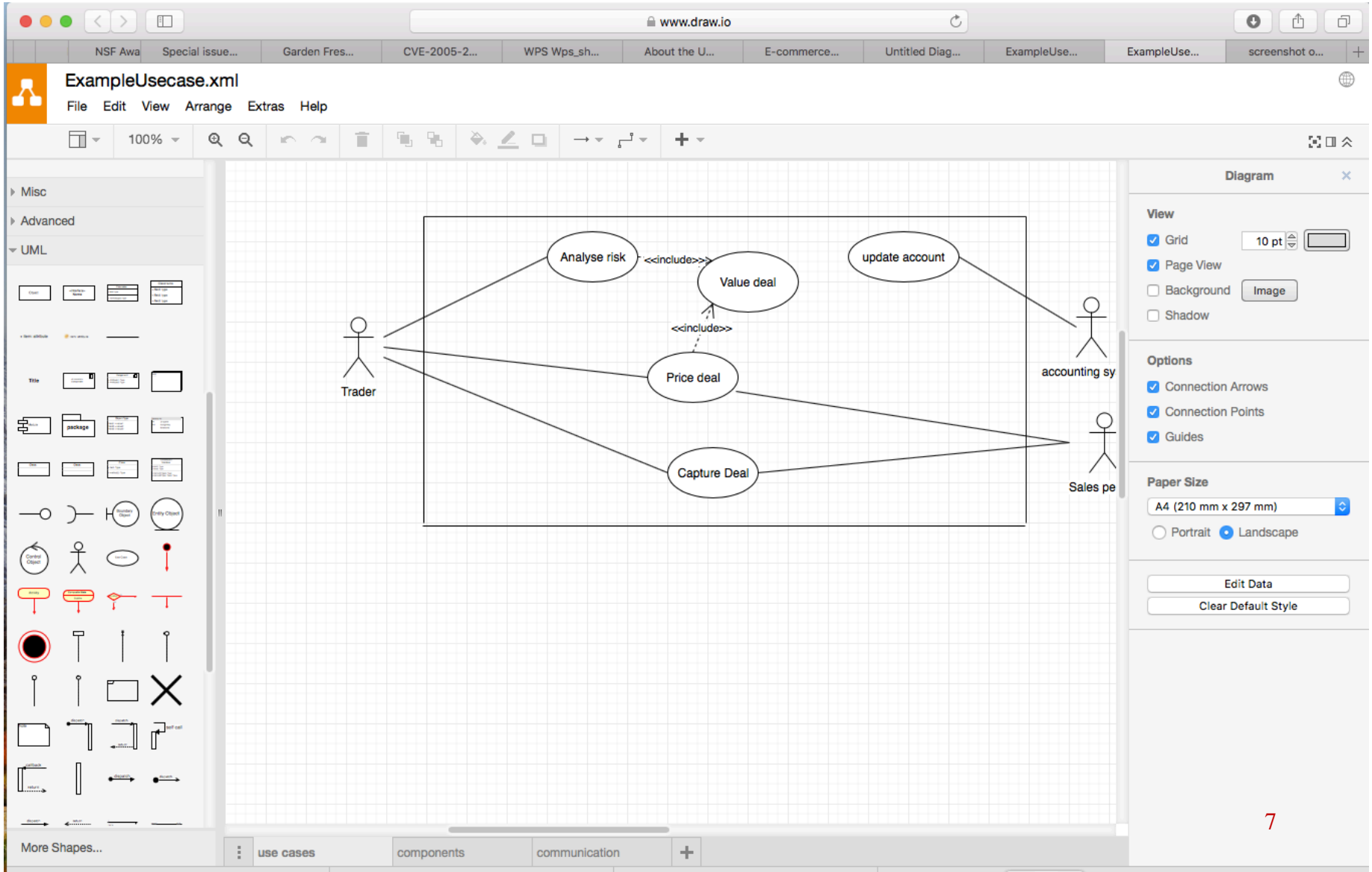
- Generalization: relationship between a general use case and a special use case 

Use Case Diagram

- Include: a chunk of behavior is similar across more than one use case.

- Extend: use case add behavior to the base use case.

Draw.io - A Tool to Design UML Diagrams



Practice - Use Case Diagram

Develop a use case diagram for the home security system.

The main functionalities are:

1. Set a system on and off
2. Collect data using sensors and camera
3. Analyze the data to detect intruders
4. Make alerts
5. Send alerts to central system
6. Keep data to assess dangerous neighborhood
7. Store structured data into an Oracle database
8. Store images in a MongoDB database

Architecture Design - Functional View

Functional view of a system defines the **architectural elements** that deliver the **functions of the system**.

Functional View

It documents the functional structure of software:

1. Key functional elements
2. Responsibilities of the elements
3. Interfaces of the elements

Functional View

Recall:

1. Include only details that have impact on the stakeholders
2. Leave details to designers

You can write all your software in one component

Or you can design it in a set of components

Component-based Style

Benefits

- Ease of deployment
- Reduced cost
- Ease of development
- Reusable
- Mitigation of technical difficulty

Component-based Style

You should consider the style to:

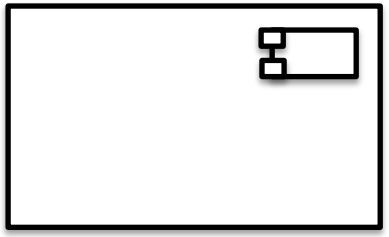
- use available third-party components
- combine components written with different programming languages
- create pluggable architecture

UML Component Diagram

- A component is a piece of software that
  - offers (via an interface) predefined services
  - has an interface to exchange data with other independent components of software
- A component could be a class or a set of classes
- Uses e.g., Remote Procedure Call
  or Message Passing

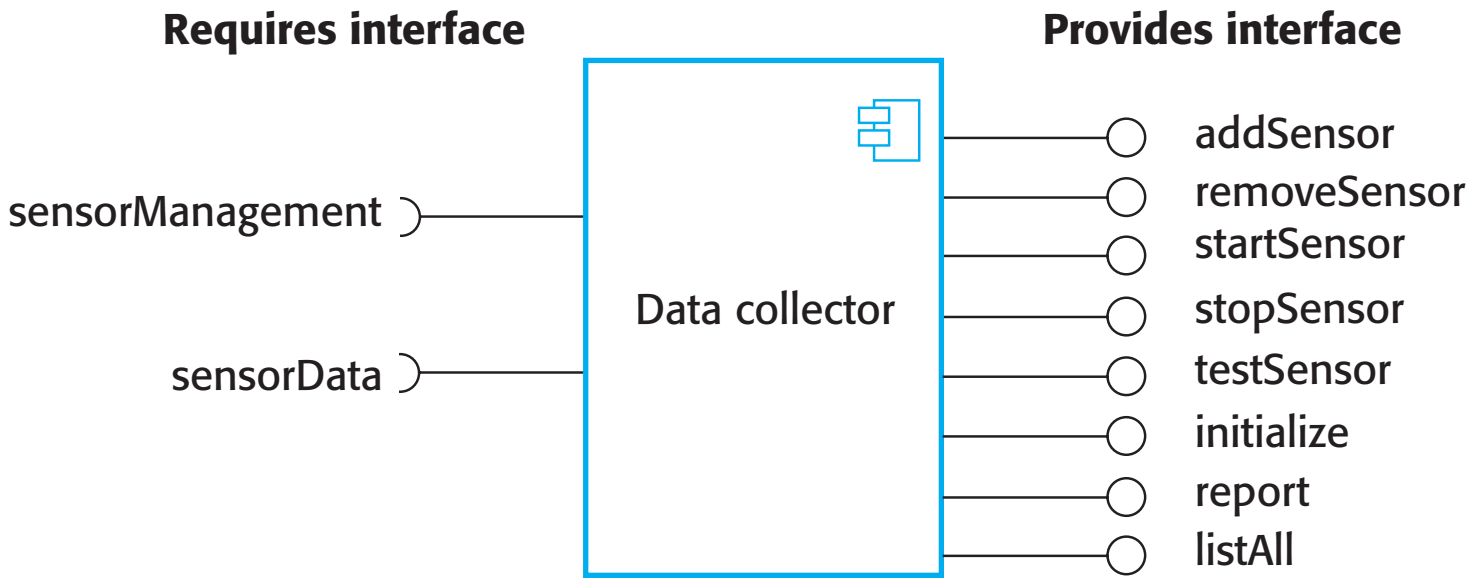
Component Diagram

- Components are independent software pieces that compose the software
  - Customers can upgrade each component separately
  - Old components can work with new components seamlessly
  - Support mix and match components of different providers

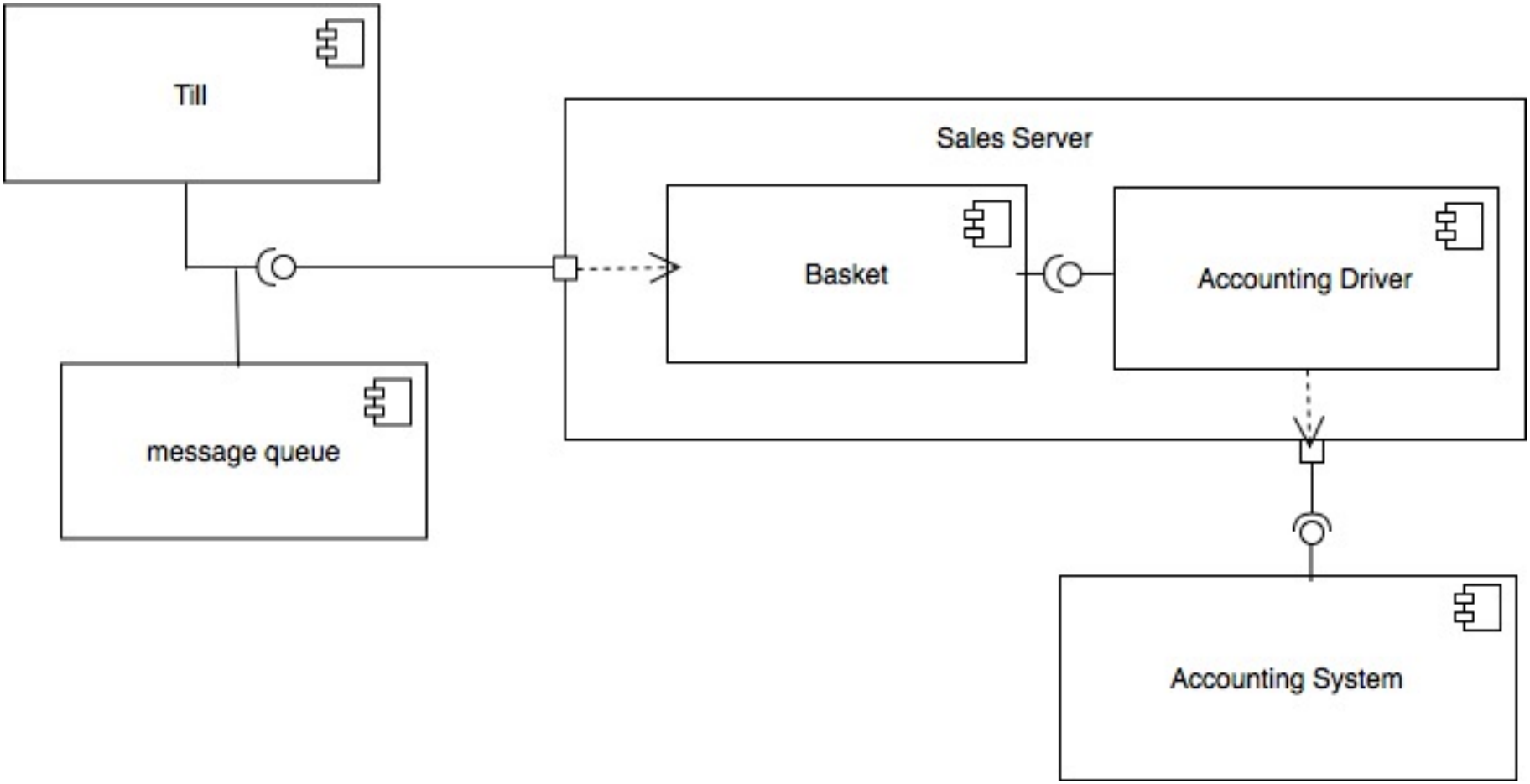


Component Interfaces

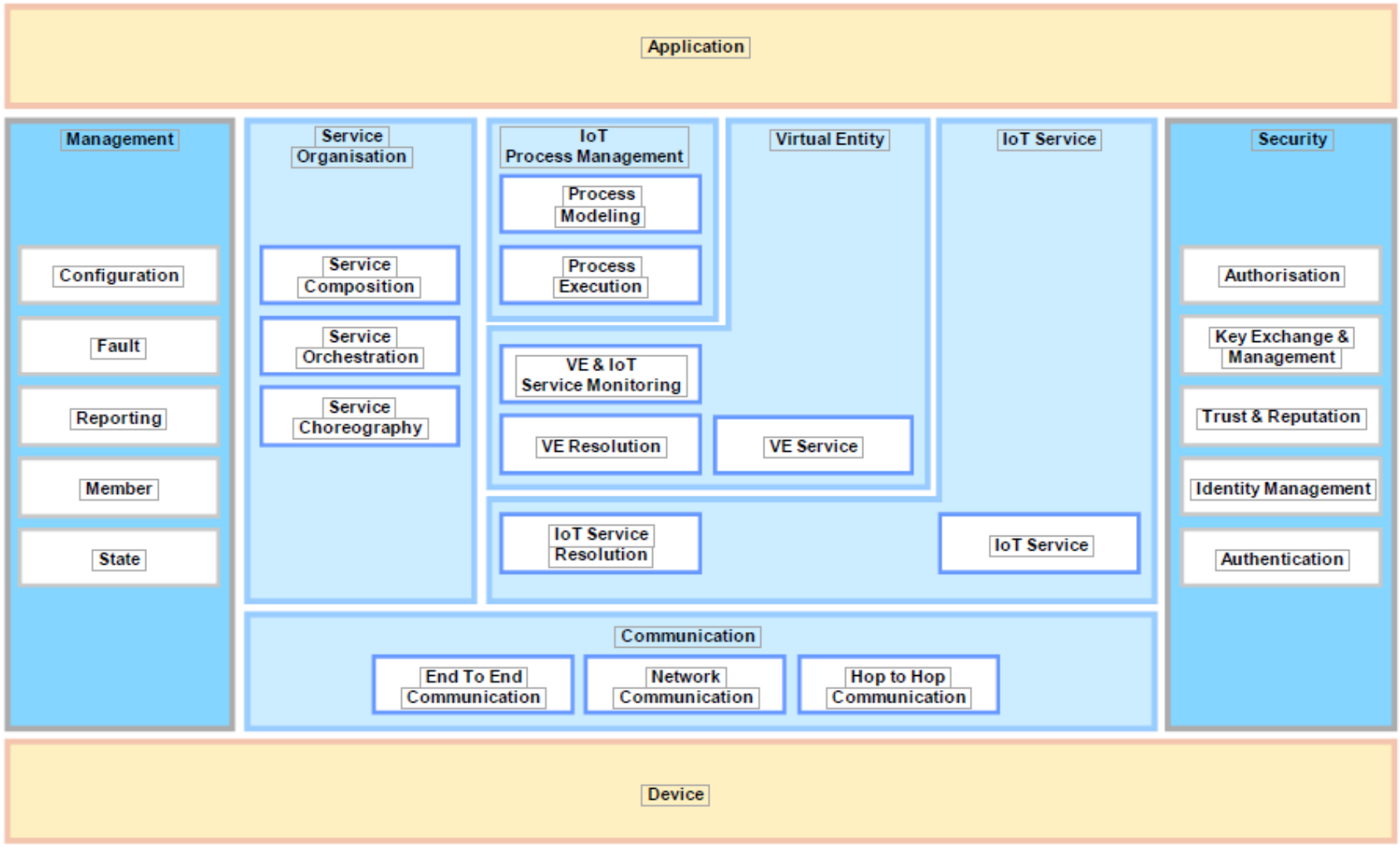
A component has an interfaces to exchange data with other independent components of software



Component Diagram



Is this a Component Diagram?



Example – Home Security System

Develop a component diagram for home security system. The system :

1. Set a system on and off
2. Collect data using sensors and camera
3. Analyze the data to detect intruders
4. Make alerts
5. Send alerts to central system
6. Keep data to assess dangerous neighborhood
7. Store structured data into an Oracle database
8. Store images in a MongoDB database

Component Diagram - Activities

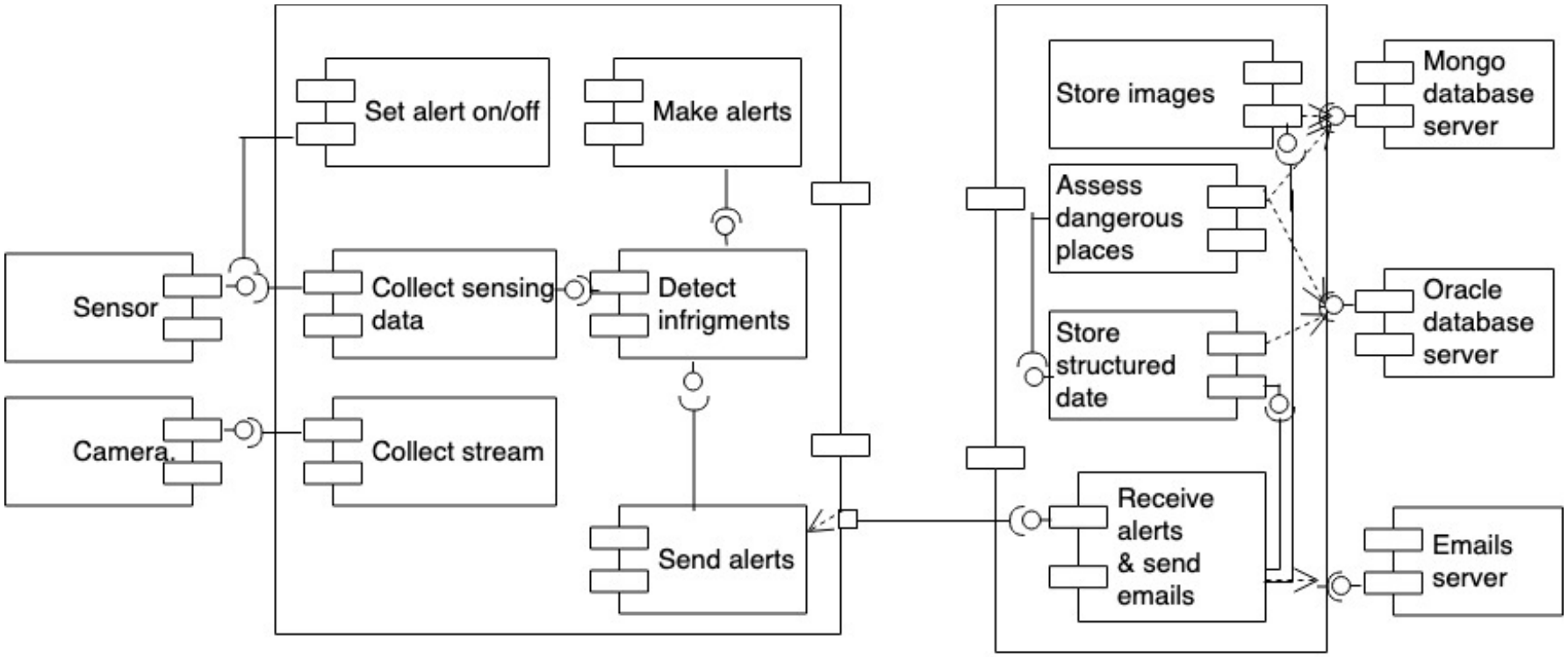
1. Identify the main components and assign responsibilities to each components
   - There are different approaches
2. Refine the components as needed
3. Design the connectors – links between the requesters and providers
4. Design the interfaces – including operations

Home Security System – Component Diagram

Develop a component diagram for home security system. The system :

1. Set a system on and off
2. Collect data using sensors and camera
3. Analyze the data to detect intruders
4. Make alerts
5. Send alerts to central system
6. Keep data to assess dangerous neighborhood
7. Store structured data into an Oracle database
8. Store images in a MongoDB database

Home Security System – Component Diagram



Home Security System

A common problem is how to group the components

⇒ Decide based on a logic/theme.

- Style: e.g., Client/server
- Third-party components often are in separate components
- Components installed on specialized devices should be in sperate components

Home Security System

Note:

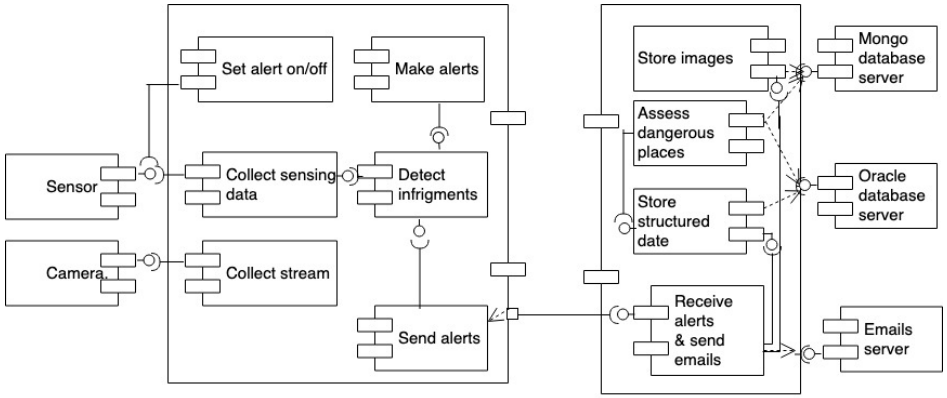
- You can distribute the software responsibilities on the components of the chosen styles, e.g., client/server

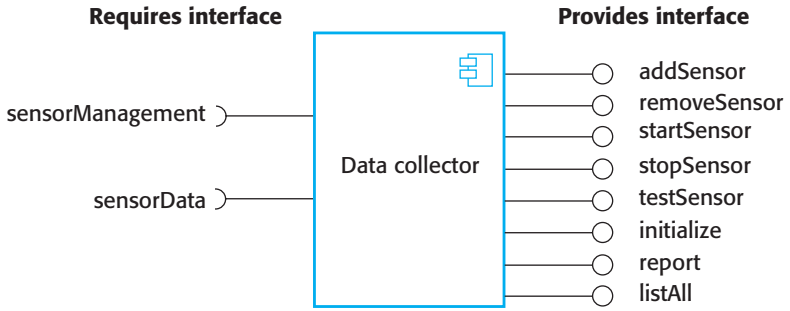
vs

- You can design your components first and then distribute them according to the chosen style

Home Security System – Component Diagram

Do we have enough details for the developers to implement the complex use cases?





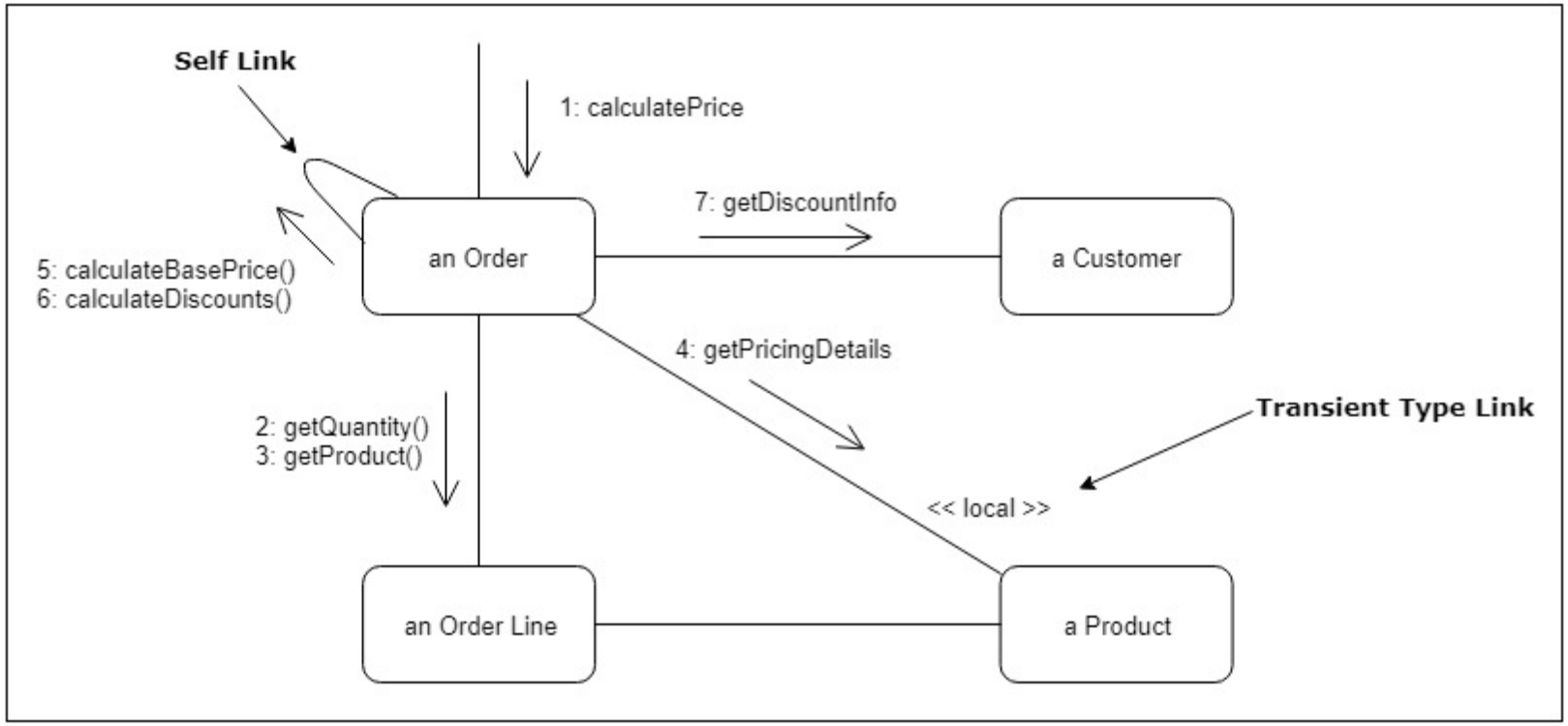
Communication Diagram

- Communication diagram shows the data links between the various participants in the interaction.

- The diagram is used to show how the components cooperate for a given use case –> It could be used to validate the component diagram

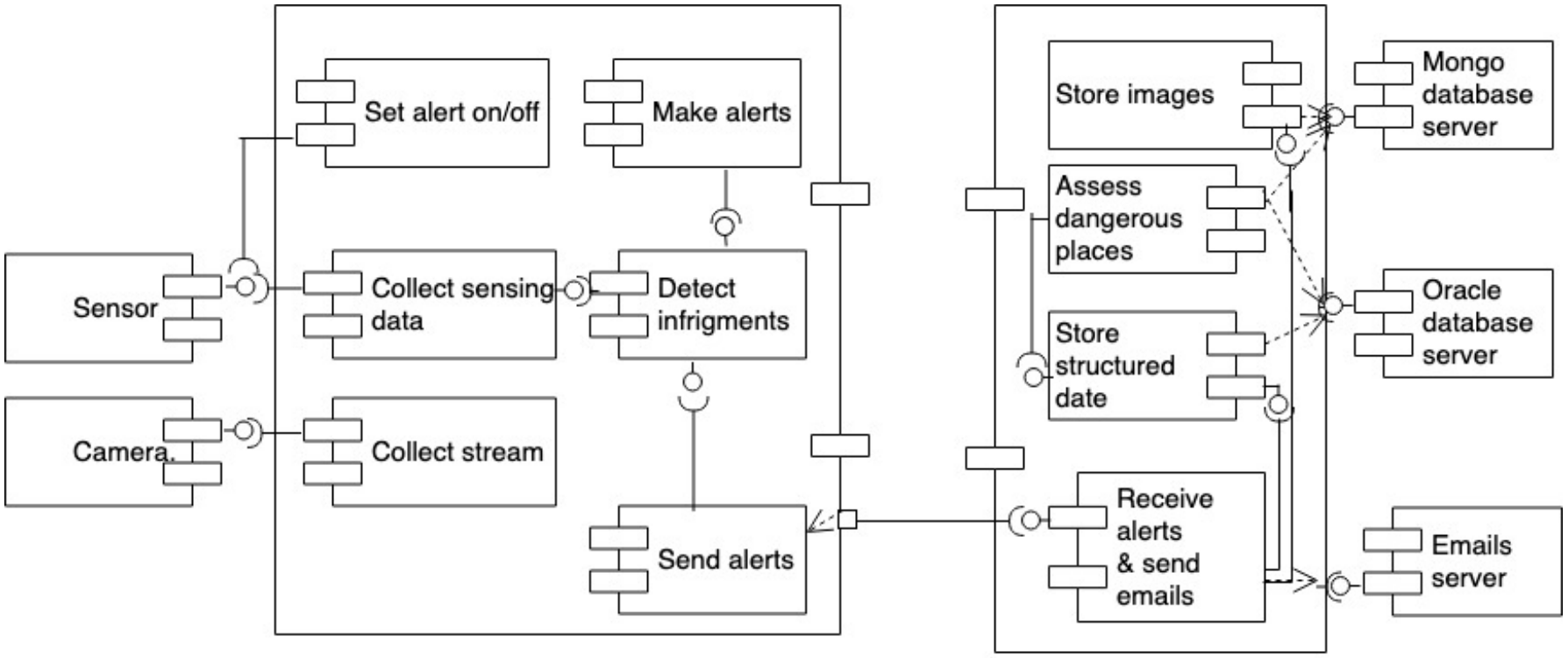
Example of Communication Diagram

- The objects are listed as rectangles and the arrows indicate the messages being passed.
- The numbers next to the messages show the sequence of the messages as they are passed between the objects



Home Security System – Communication Diagram

Develop a communication diagram for make alerts use case



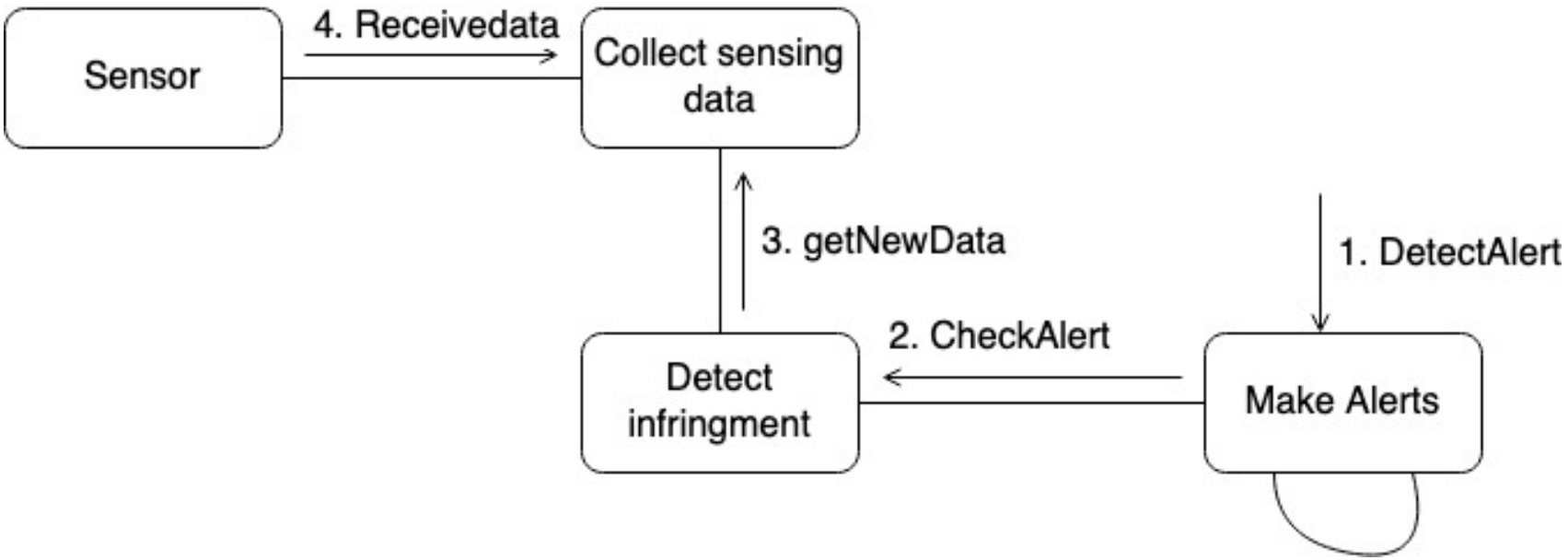
Home Security System – Communication Diagram

- Two scenarios for **make alerts** use case
  1. The *Make alert* applies periodically the algorithm *detect infringement* on data received from the *sensor*
  2. The algorithm *detect infringement* runs on incoming *data* and sends a notification to *make alerts* component when there is an infringement
  3. You may have another idea

Implement the method as a sequence of main **calls on methods** on the **participating components**

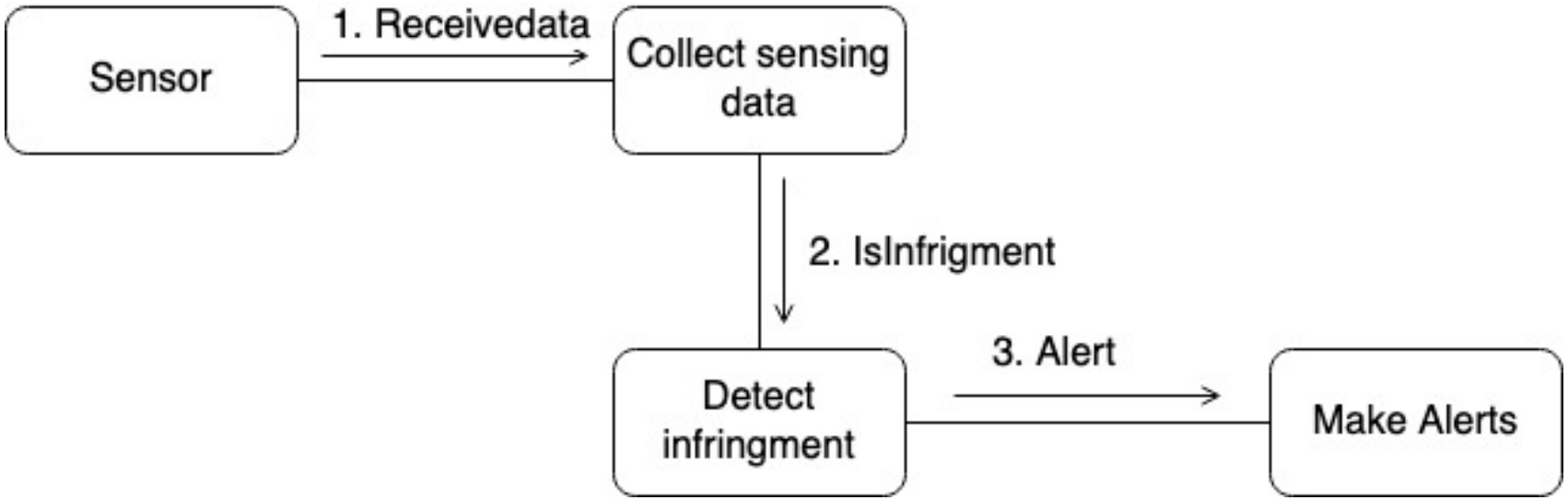
Home Security System – Communication Diagram

Scenario 1



Home Security System – Communication Diagram

Scenario 2



Verify an Architecture

Check consistency of the provided models:

1. Use cases are supported by the communication diagrams
2. All the components are in the communication diagrams
3. You may have other ideas

Participation 4

Description: The customer browses the catalog and adds desired items to the shopping basket. When the customer wishes to pay, they describe the shipping and credit card information and confirm the sale. The system checks the authorization on the credit card and confirms the sale both immediately and with a follow-up e-mail.

Work: Develop

1- Use case diagram

2- Component diagram

3- Communication diagrams for the complex use cases

Thank you

Next topic: Architecture for Single-Process Systems