Overview of Software Architecture

Dr. Lotfi ben Othmane
University of North Texas

Definition

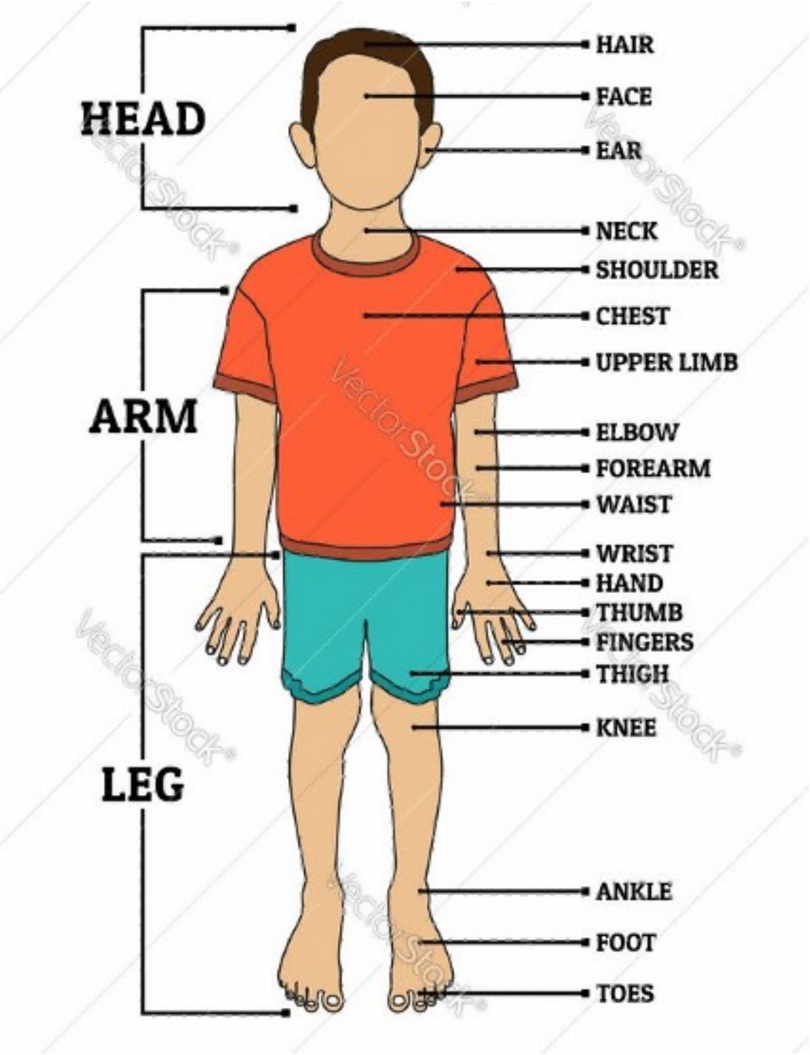
The software architecture of a program or computing system is the structure or structures of the system, which comprise software elements, the externally visible properties of those elements, and the relationships among them.

L.Bass, P.Clements, R.Kazman,
Software Architecture in Practice (2nd edition), Addison-Wesley 2003

Structure – Human

Goal: Represent the internal structure of the system

- A system has a set of functions
  - Move
  - Eat
  - Speak
  - Smile

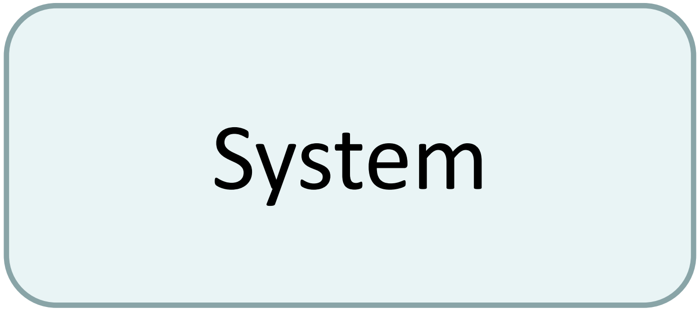


Context Diagram

Want a boxing Robot

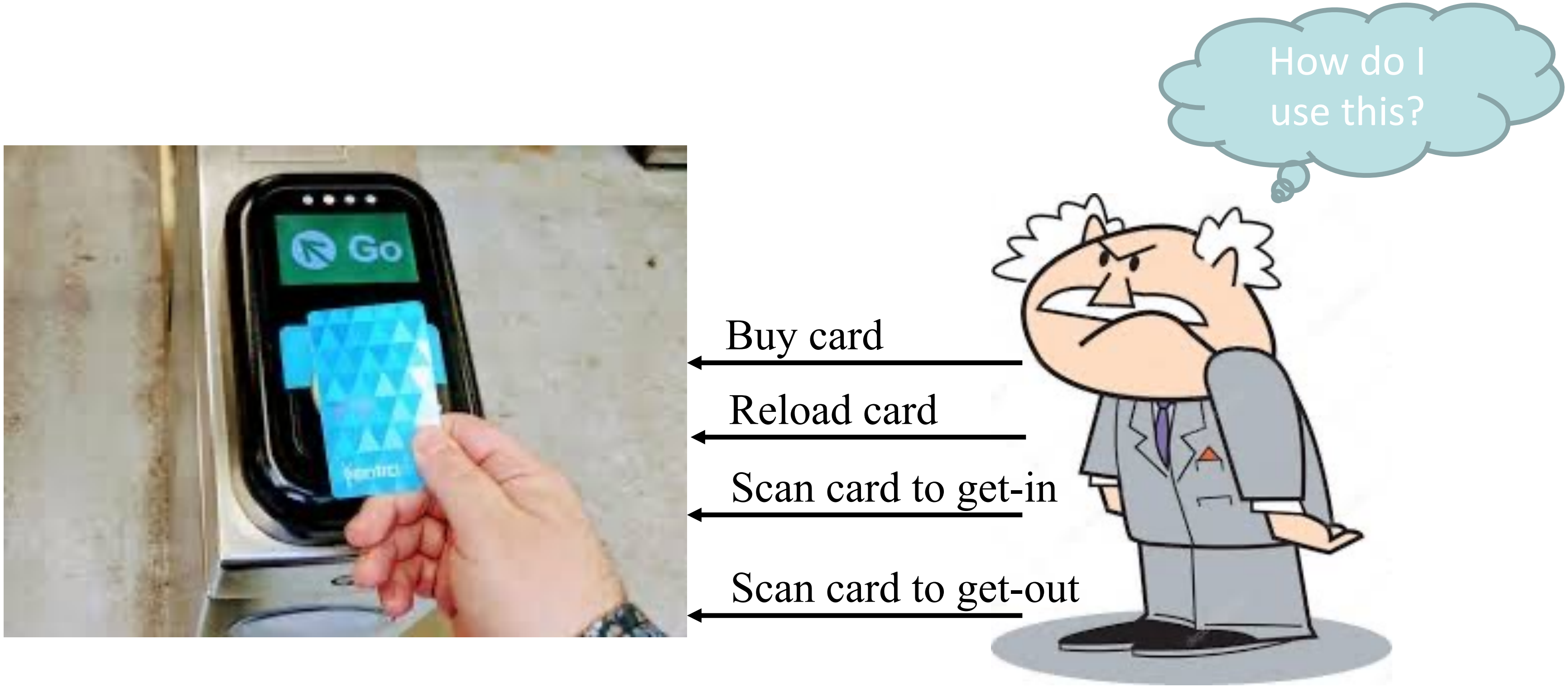
First, What is the Context?

one box


System

Context

Helping Mr. Grandy



How do I use this?

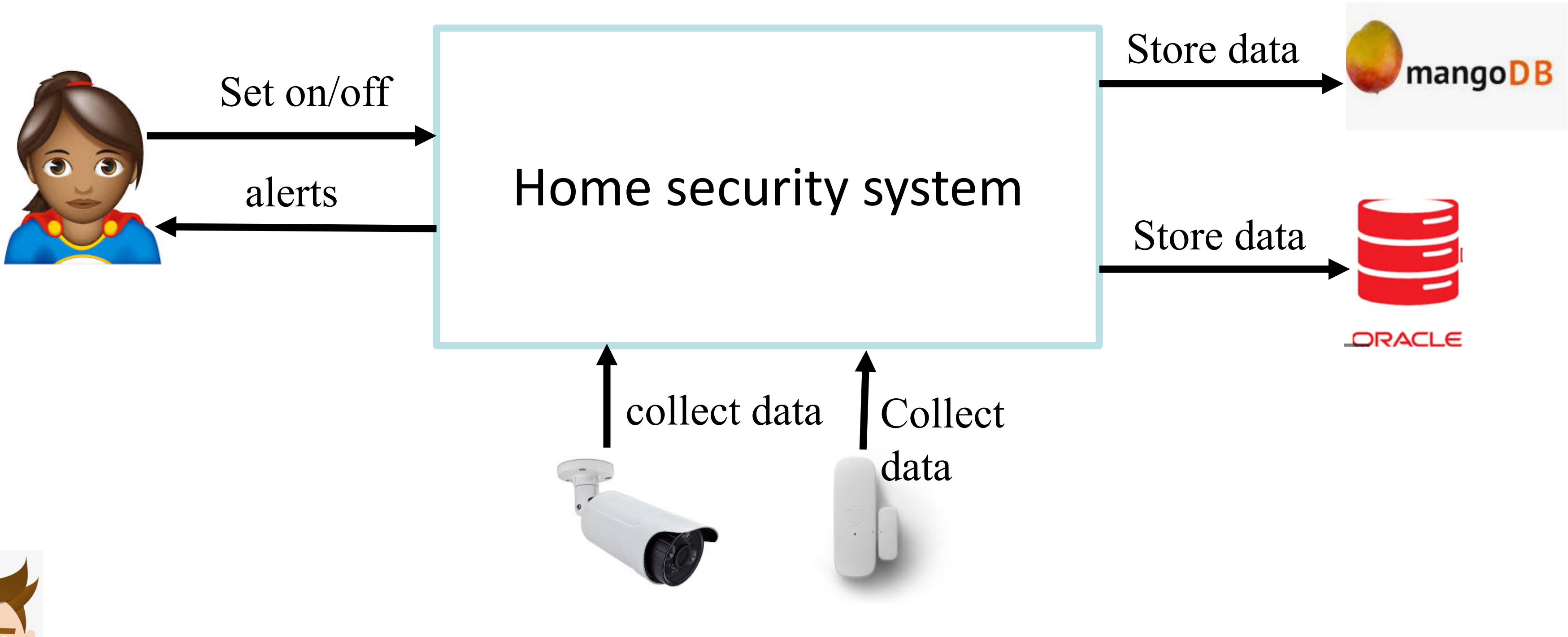
Buy card

Reload card

Scan card to get-in

Scan card to get-out

Context Diagram – Home Security System



Set on/off

alerts

Home security system

Store data

Store data

collect data

Collect data

Should the camera
1- stream video all the time?
2- stream when requested?
3- stream when it detects movement?

Exercise: System Scope and Responsibilities

Design a context diagram for the system

Example – Online retail system

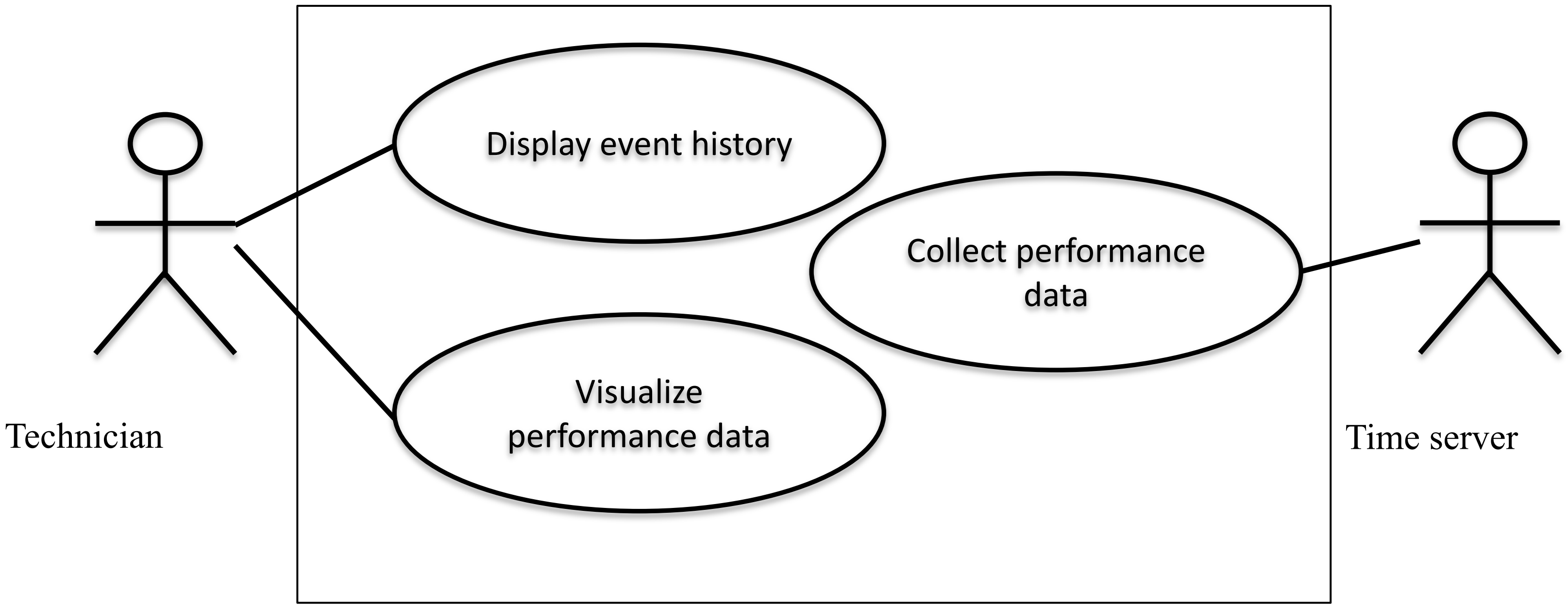
1. Present the products and services catalog to the user
2. Provide a flexible search capability
3. Accept orders of goods
4. Accept payments by credit card
5. Provide support for fulfillment

Exclusion:

1. Amend or cancel orders
2. Show inventory

Primary Functionality

We use UML use case diagram to represent the functionalities of a system

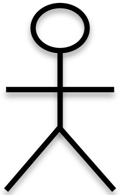


Use Case Diagram

- System boundary: Rectangle shape representing the boundary of the system.

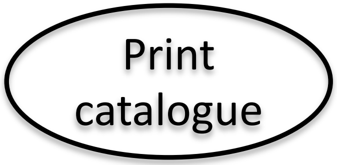
- Actors: A role that a user plays with respect to the system

  An actor could be a system



Actor

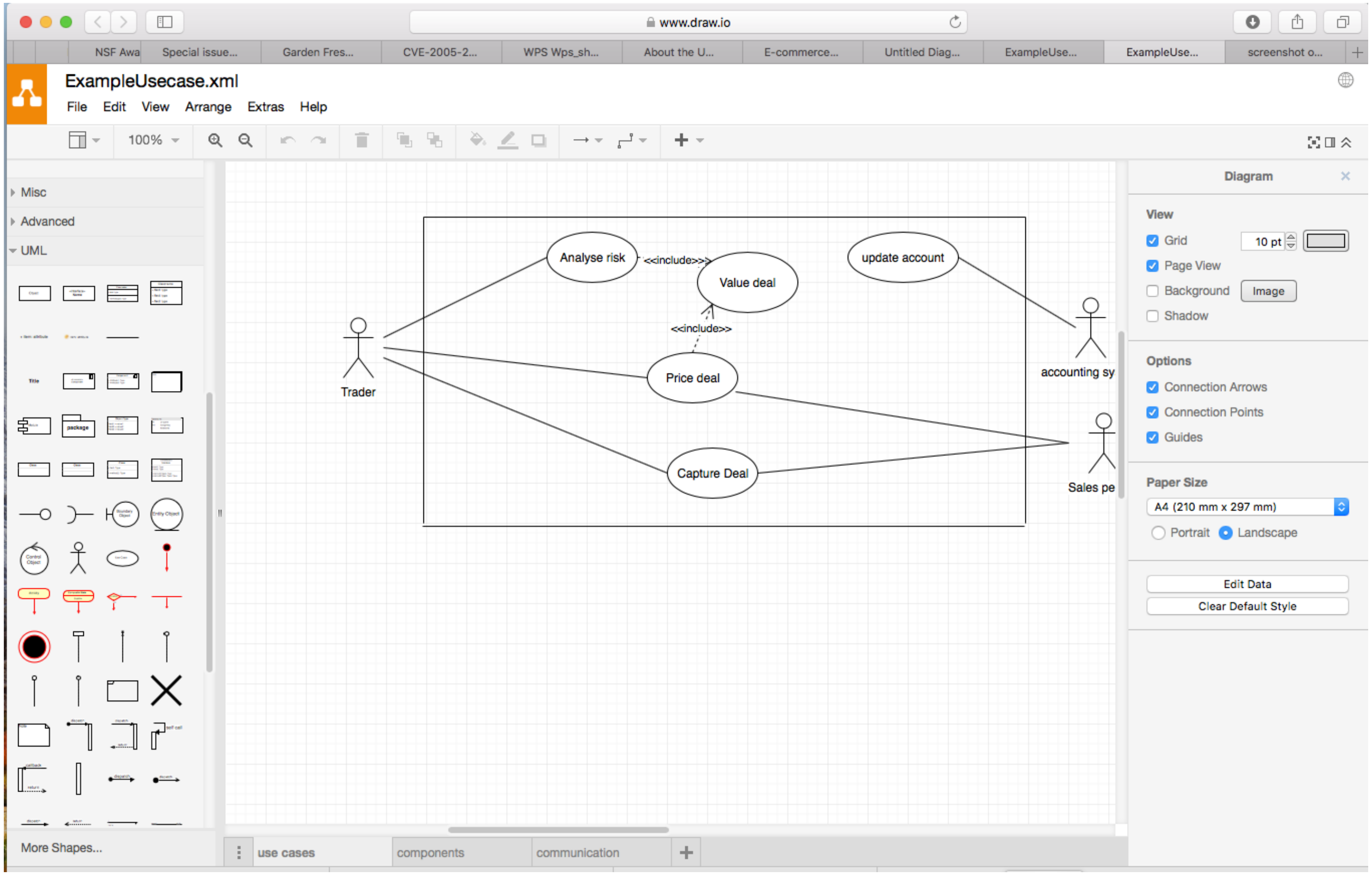
- Use case: A set of scenarios that describe the interactions of the actors with the system



Use case

Draw.io - A Tool to Design UML Diagrams

https://app.diagrams.net



ExampleUsecase.xml

Diagram

View
- ☑ Grid 10 pt
- ☑ Page View
- ☐ Background
- ☐ Shadow

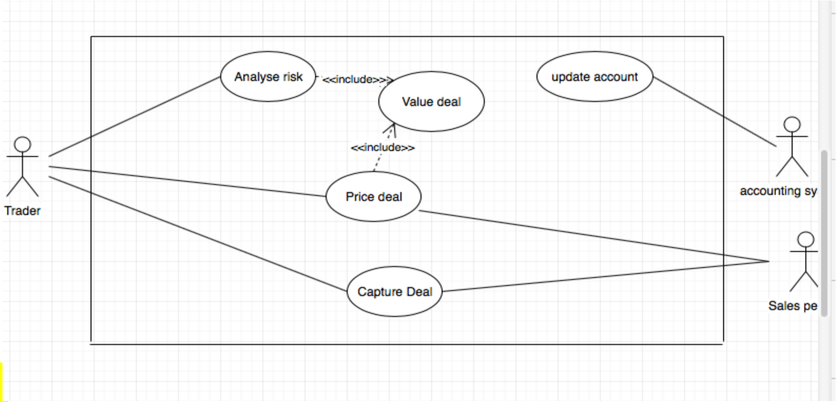
Options
- ☑ Connection Arrows
- ☑ Connection Points
- ☑ Guides

Paper Size
A4 (210 mm x 297 mm)
○ Portrait ● Landscape

Exercise - Use Case Diagram

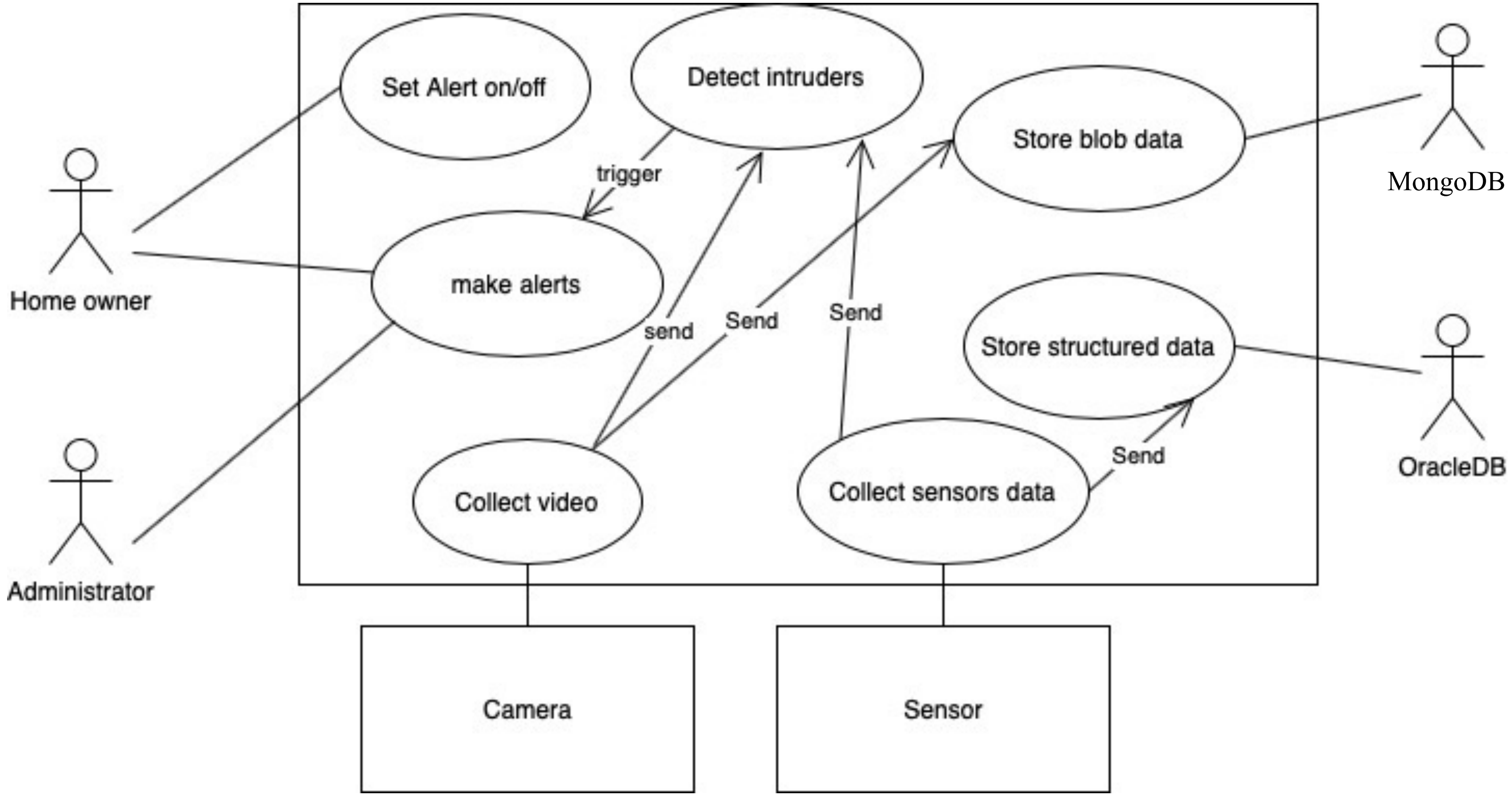
Develop a use case diagram for the home security system.

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



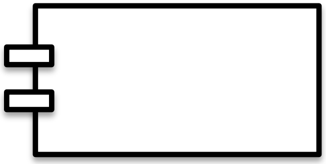
Home Security System – Use Case Diagram

Could be improved.

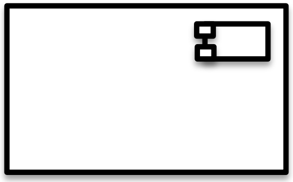


Component Diagram

Components are independent software pieces that compose the software

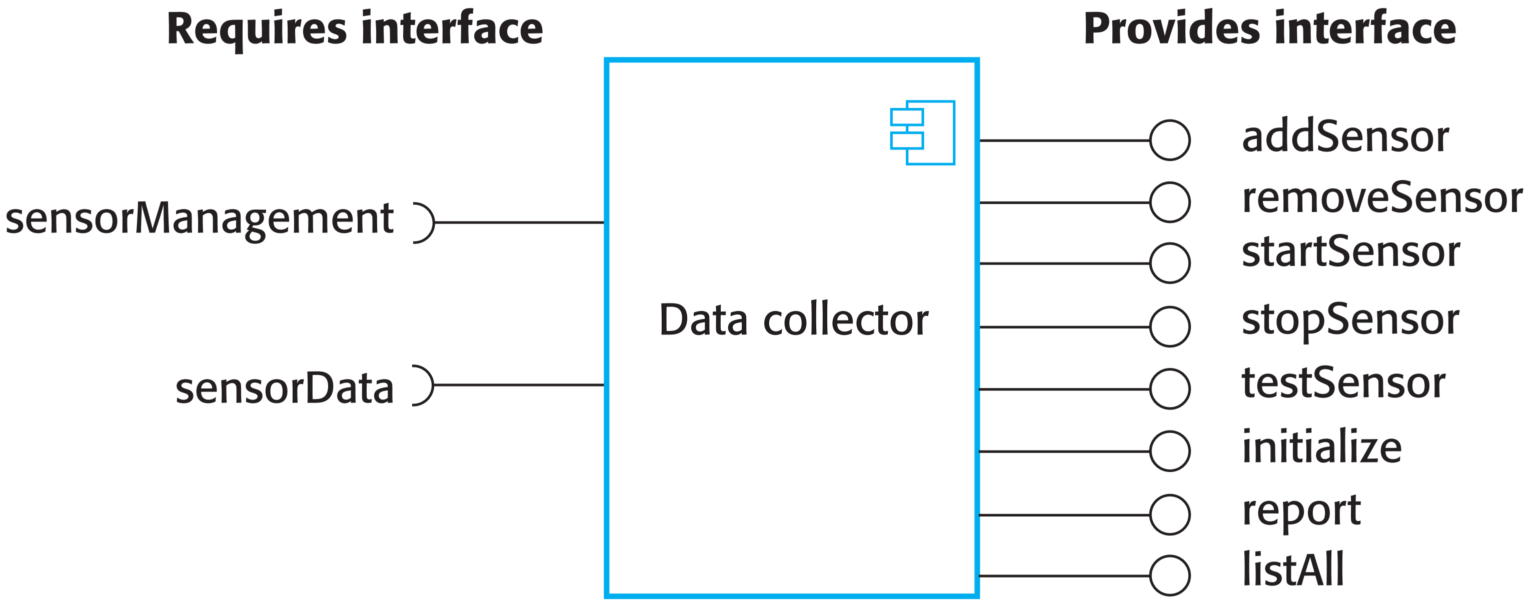


1. Customers can upgrade each component separately
2. Old components can work with new components seamlessly
3. Support mix and match components of different providers

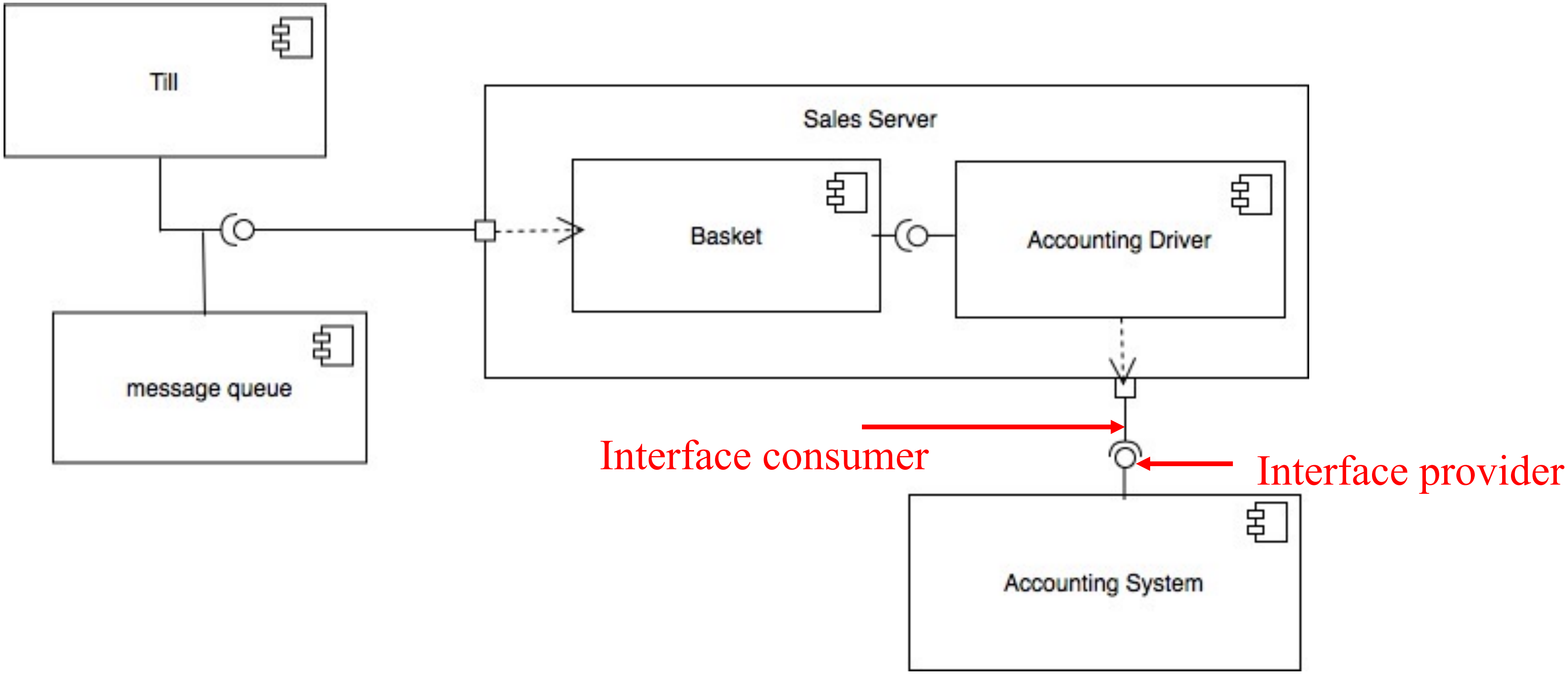


Component Interfaces

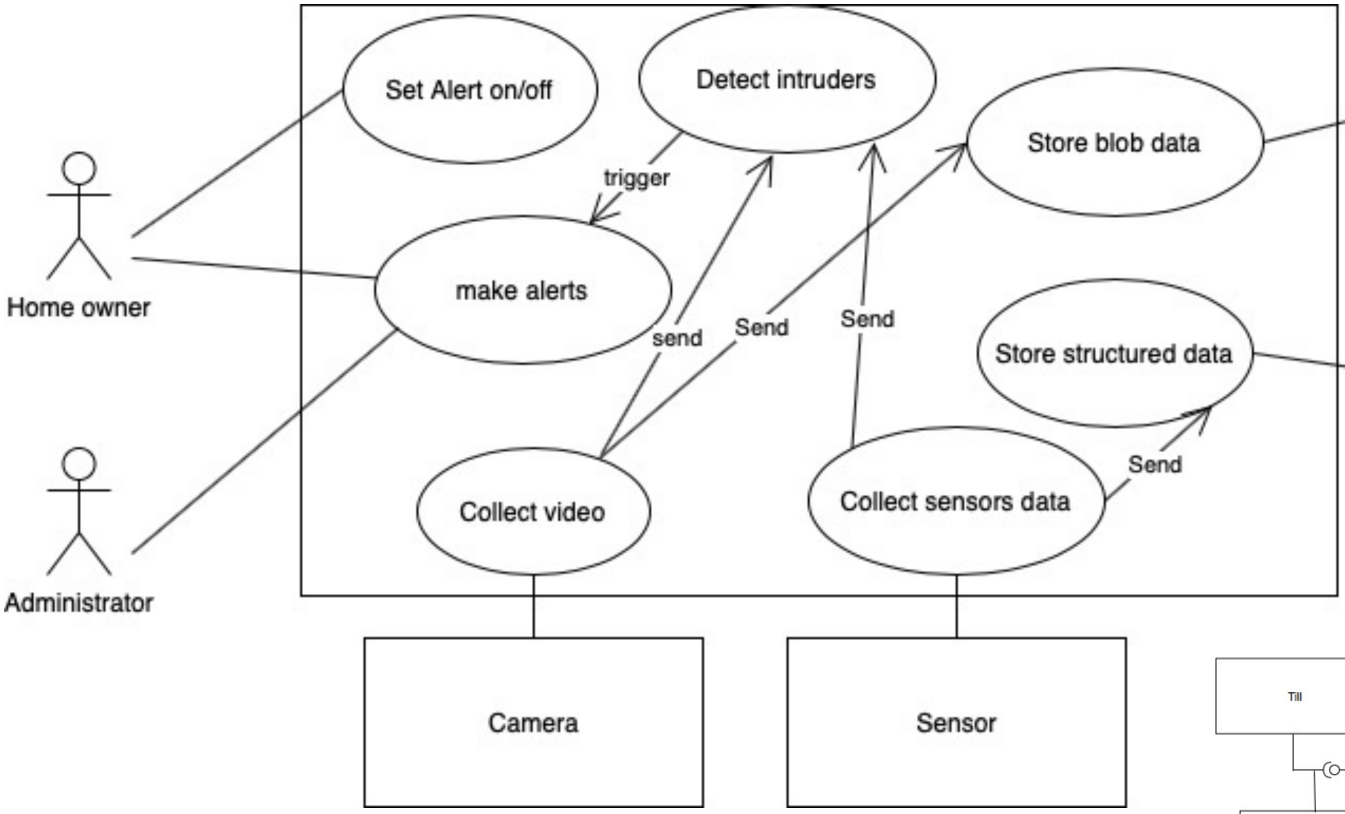
A component has an interface to exchange data with other independent components of software



UML Component Diagram

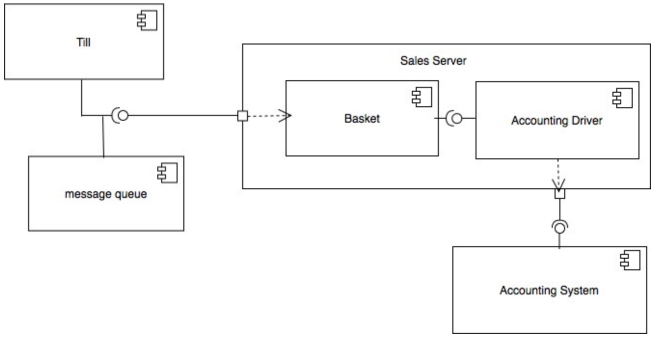


Exercise: Develop a Component Diagram For the Security System



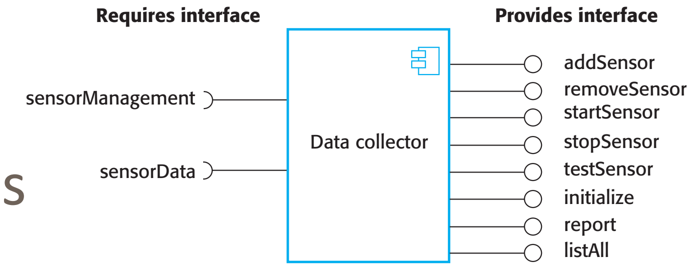
Start the component diagram for the Home Security System

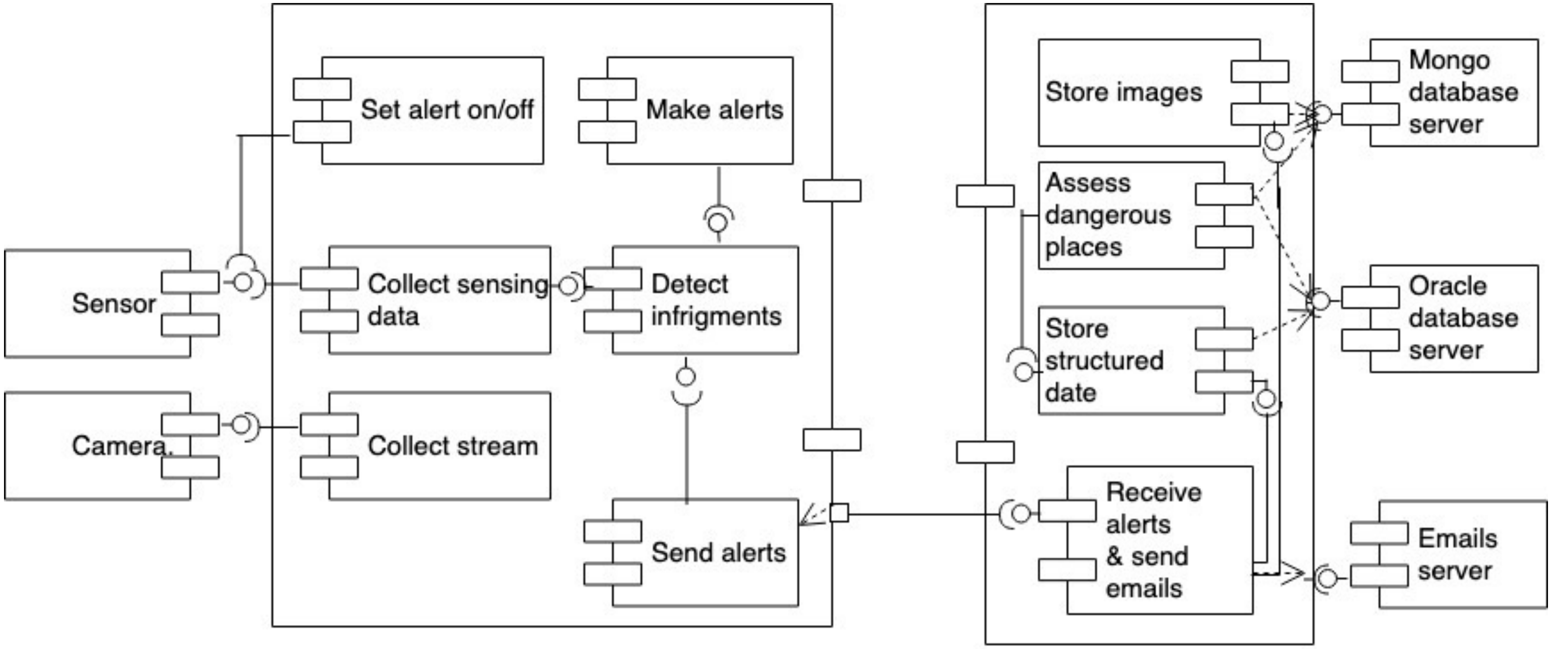
http://draw.io



Home Security System – Component Diagram

We need the interfaces of the components



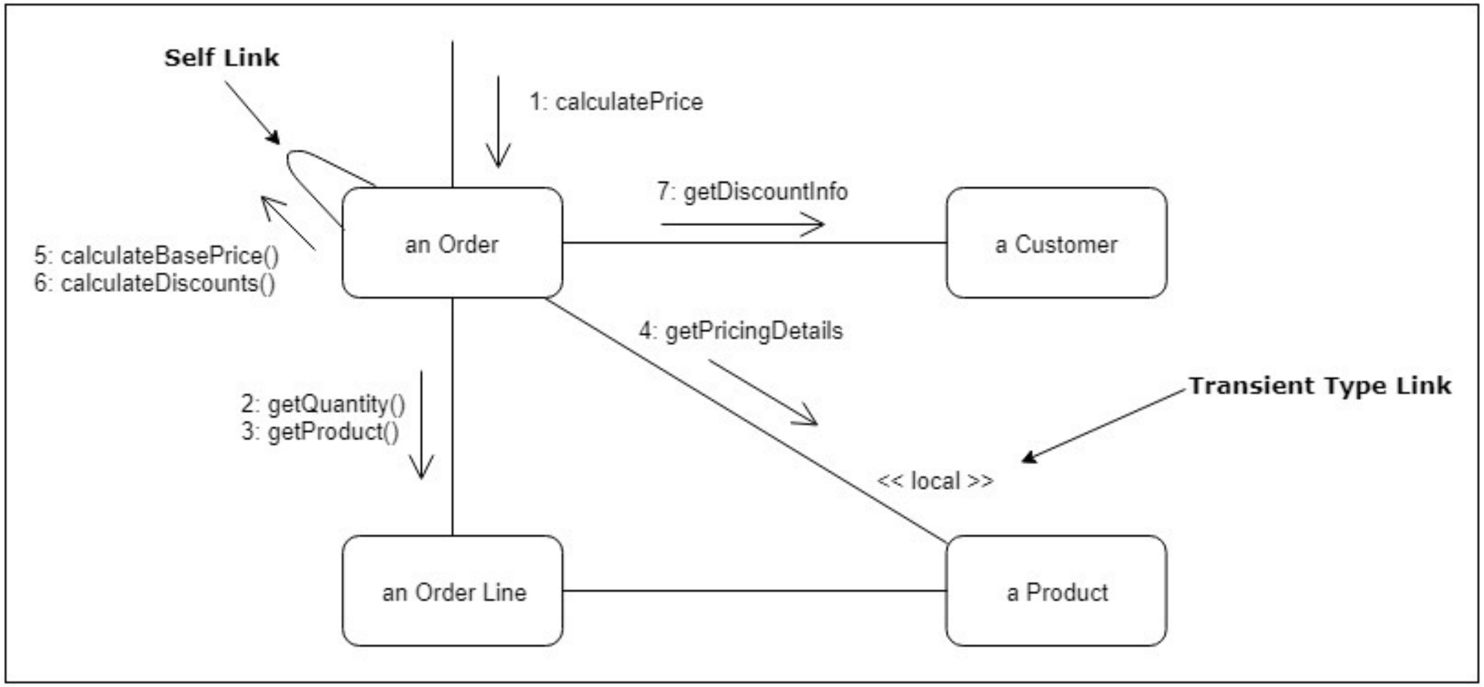


- Communication diagram shows the data links between the various participants in the system.

- 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

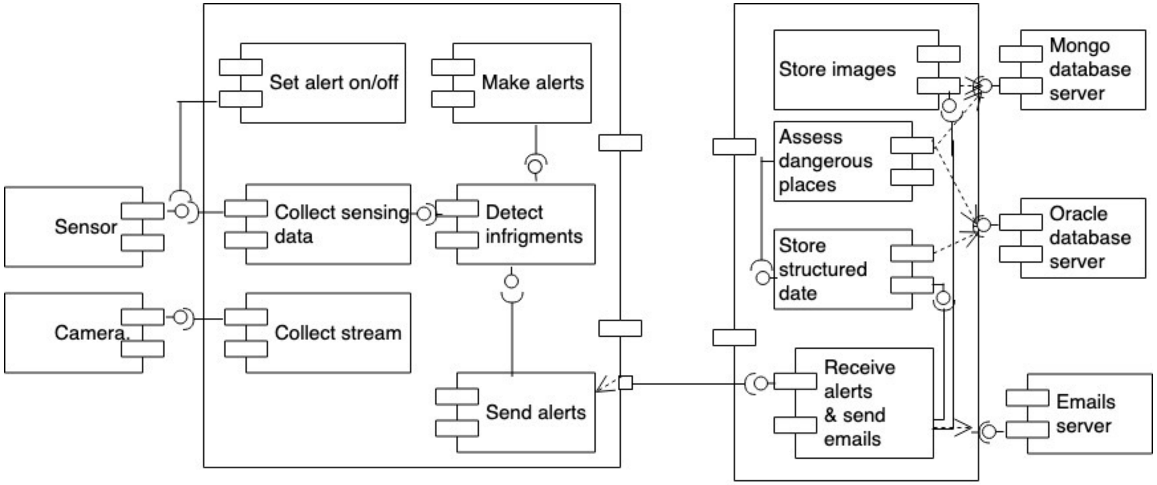
- The components are listed as rectangles and the arrows indicate the messages being passed.
- The numbers next to the messages show the sequence of the messages as they are passed between the components

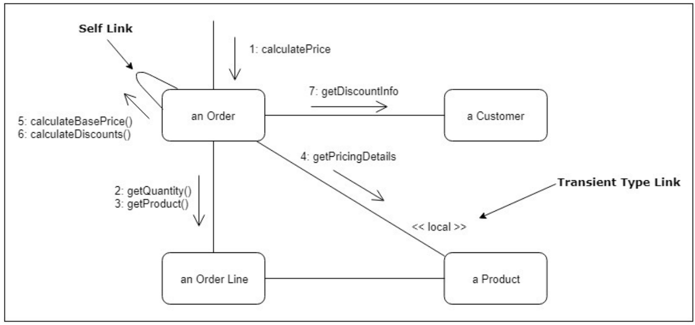


Home Security System – Communication Diagram

make alerts use case

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





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*

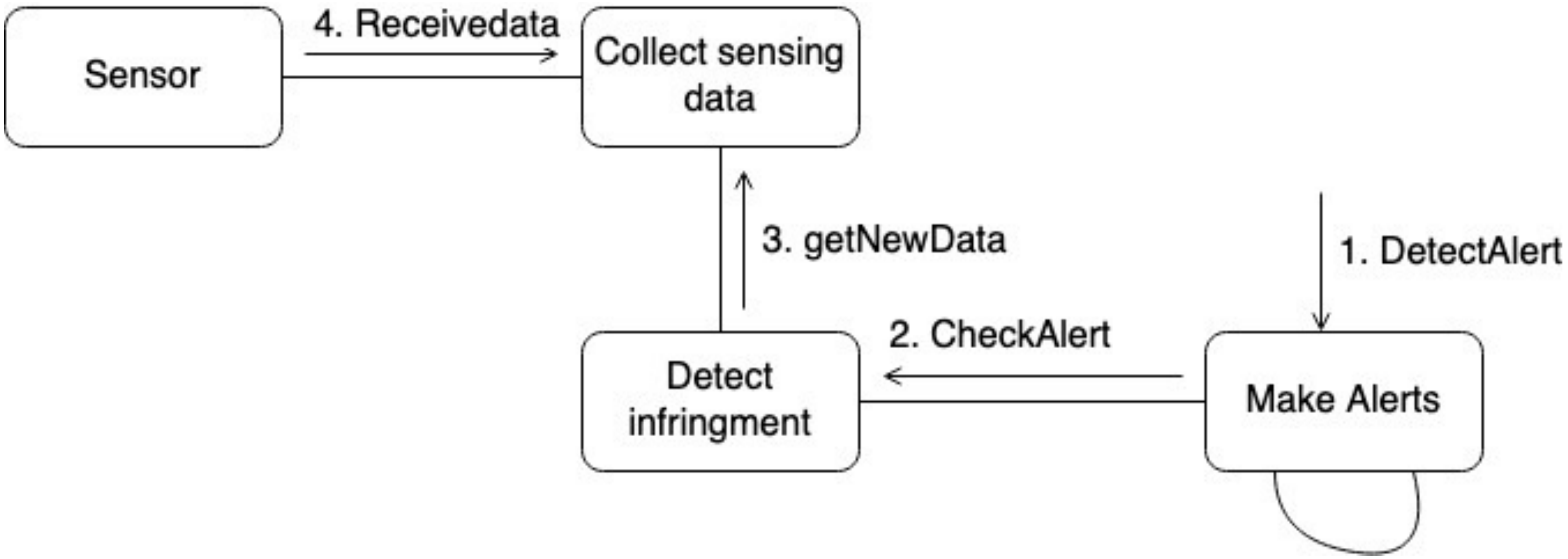
2. The algorithm *detect infringement* runs on incoming *data* and sends a notification to *make alerts* component when there is an infringement

You may have another idea

Home Security System – Communication Diagram

Scenario 1

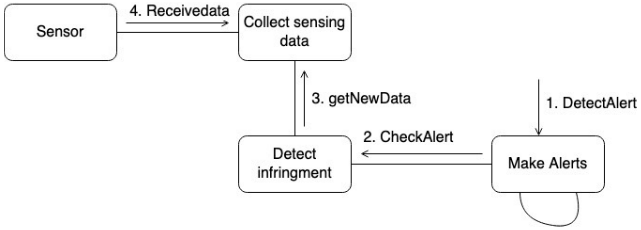
The use case calls method 1

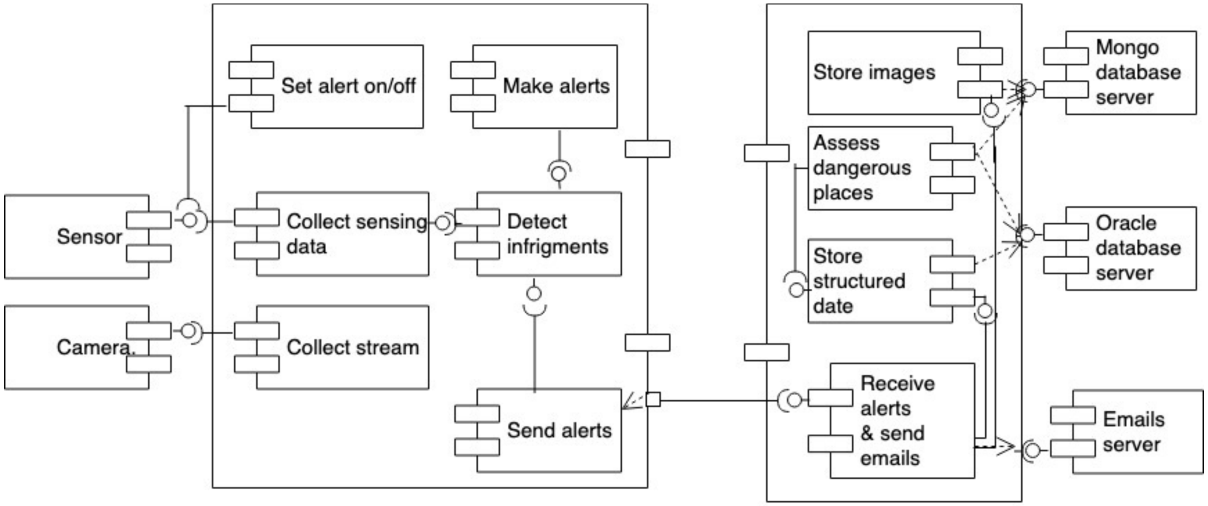


Exercise - Communication Diagram

Develop the communication diagram for scenario 2.

The algorithm *detect infringement* runs on incoming *data* and sends a notification to *make alerts* component when there is an infringement

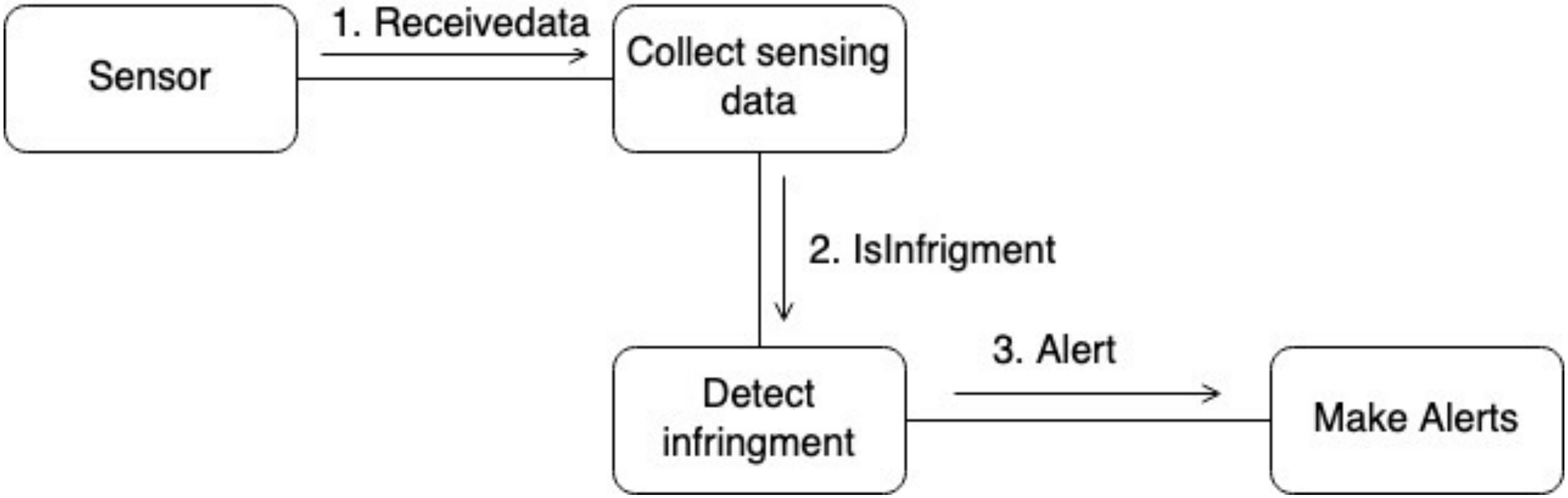




Home Security System – Communication Diagram

Scenario 2

The use case calls method 1



Validate an Architecture

Check consistency of the provided models:

1. All use cases could be implemented using the identified components
2. Each of the components participates in at least one use case
3. The complex use cases are supported by the communication diagrams
4. All the components of the communication diagrams are in the component diagram
5. You may have other ideas

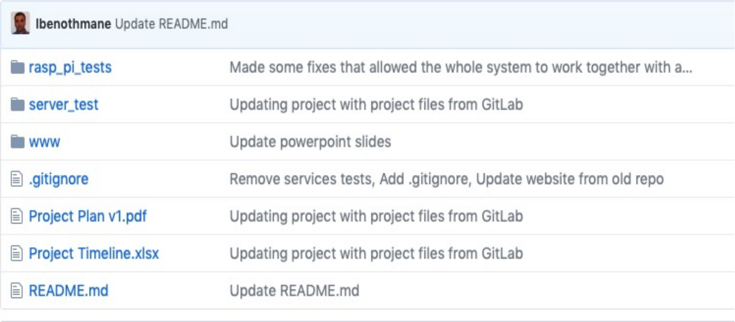
Architecture Recovery

- Architecture recovery is the extraction and analysis of a software architecture

- Current tools cluster the software code into packages

Architecture Recovery From Code

- Simple methods – Use the folders' structure
- Dependency-based methods – Use function calls network
- Text-based methods – Mine the text code



lbenothmane Update README.md

rasp_pi_tests	Made some fixes that allowed the whole system to work together with a...
server_test	Updating project with project files from GitLab
www	Update powerpoint slides
.gitignore	Remove services tests, Add .gitignore, Update website from old repo
Project Plan v1.pdf	Updating project with project files from GitLab
Project Timeline.xlsx	Updating project with project files from GitLab
README.md	Update README.md

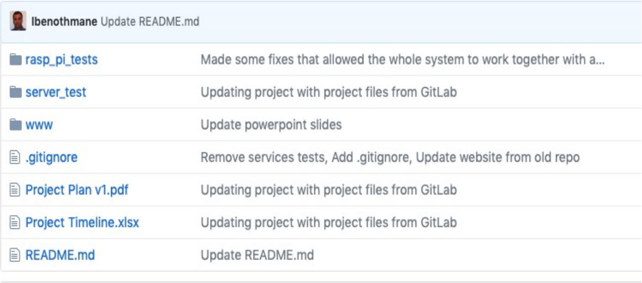
This is not an exhaustive list

Architecture Recovery By Observing The Project Folders and Files

- The folders are supposed to show grouping of functionalities

→ They indicate the components of the system

- Architecture includes program files, configuration files, page/form templates, etc.

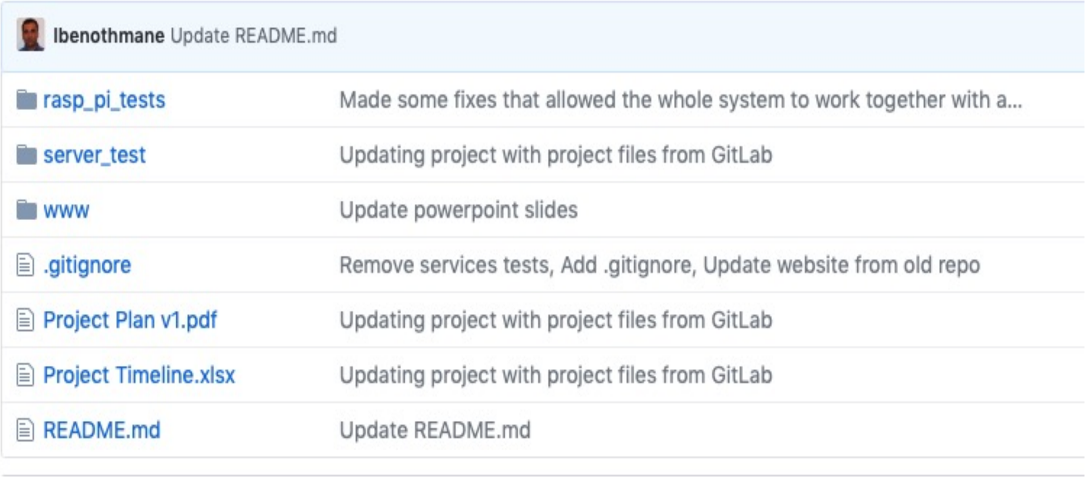


lbenothmane Update README.md

rasp_pi_tests	Made some fixes that allowed the whole system to work together with a...
server_test	Updating project with project files from GitLab
www	Update powerpoint slides
.gitignore	Remove services tests, Add .gitignore, Update website from old repo
Project Plan v1.pdf	Updating project with project files from GitLab
Project Timeline.xlsx	Updating project with project files from GitLab
README.md	Update README.md

Exercise - Architecture Recovery

- Try to identify the components for each process—get one more level of details.
- Draw a UML component diagram of the system.
- The link is: https://github.com/lbenothmane/FleetManagement

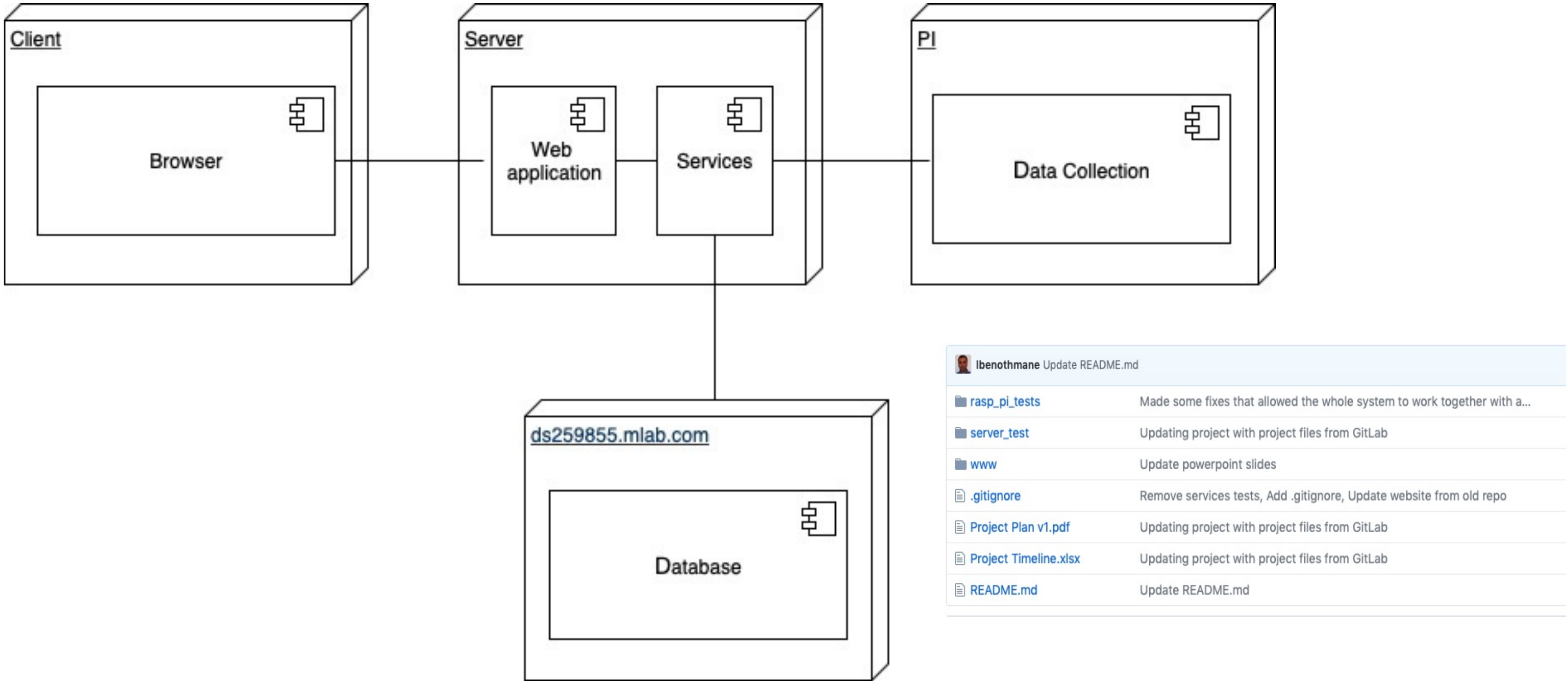


lbenothmane Update README.md

rasp_pi_tests	Made some fixes that allowed the whole system to work together with a...
server_test	Updating project with project files from GitLab
www	Update powerpoint slides
.gitignore	Remove services tests, Add .gitignore, Update website from old repo
Project Plan v1.pdf	Updating project with project files from GitLab
Project Timeline.xlsx	Updating project with project files from GitLab
README.md	Update README.md

Architecture Recovery By Observing Software Execution

You need to run the system and observe the processes.



lbenothmane Update README.md

rasp_pi_tests	Made some fixes that allowed the whole system to work together with a...
server_test	Updating project with project files from GitLab
www	Update powerpoint slides
.gitignore	Remove services tests, Add .gitignore, Update website from old repo
Project Plan v1.pdf	Updating project with project files from GitLab
Project Timeline.xlsx	Updating project with project files from GitLab
README.md	Update README.md

Conclusion

The software architecture of a program or computing system is the structure or structures of the system, which comprise software elements, the externally visible properties of those elements, and the relationships among them.

L.Bass, P.Clements, R.Kazman, Software Architecture in Practice (2nd edition), Addison-Wesley 2003

Thank you

Any Question?