

# LabQakExamples2021 - Case Study - es3

## Introduction

## Requirements

**es3-** Write a service able to answer to requests sent by a remote clients by asking the client - if it is the case - for some missing information

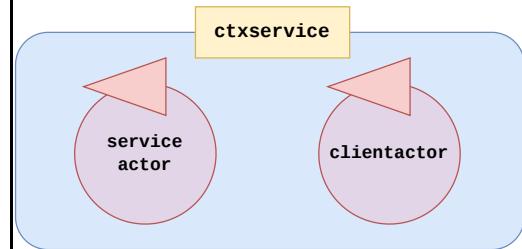
## Requirement analysis

The consumer requires to design a simple system (using the **QA-System** given by the same consumer, see [QAK21Intro](#)) that is a service able to answer to requests sent by remote clients. **Consumer specify nothing about the number of the clients or the content of the message** but he says that some messages should not contain all informations needed to the service in order to answer. The consumer also uses the words **if it is the case** but newly **specify nothing about when is the case** to ask missing information so we use a *random boolean*. In addition to this, **consumer not specify if a response containing the additional informations is really completed** or if there are others missing informations: then, we decide that **the service always check all messages delivered by the client** (see the Problem Analysis).

According to the tools provided by the **QA-System**, in order to meet the requirements and realize the new system, we need to introduce two actors:

- **serviceactor** that is the actor that realizes the answering service;
- **clientactor** that is an actor that simulates the client that sends requests (it is not needed to satisfy the requirements but is useful to simulate the interaction).

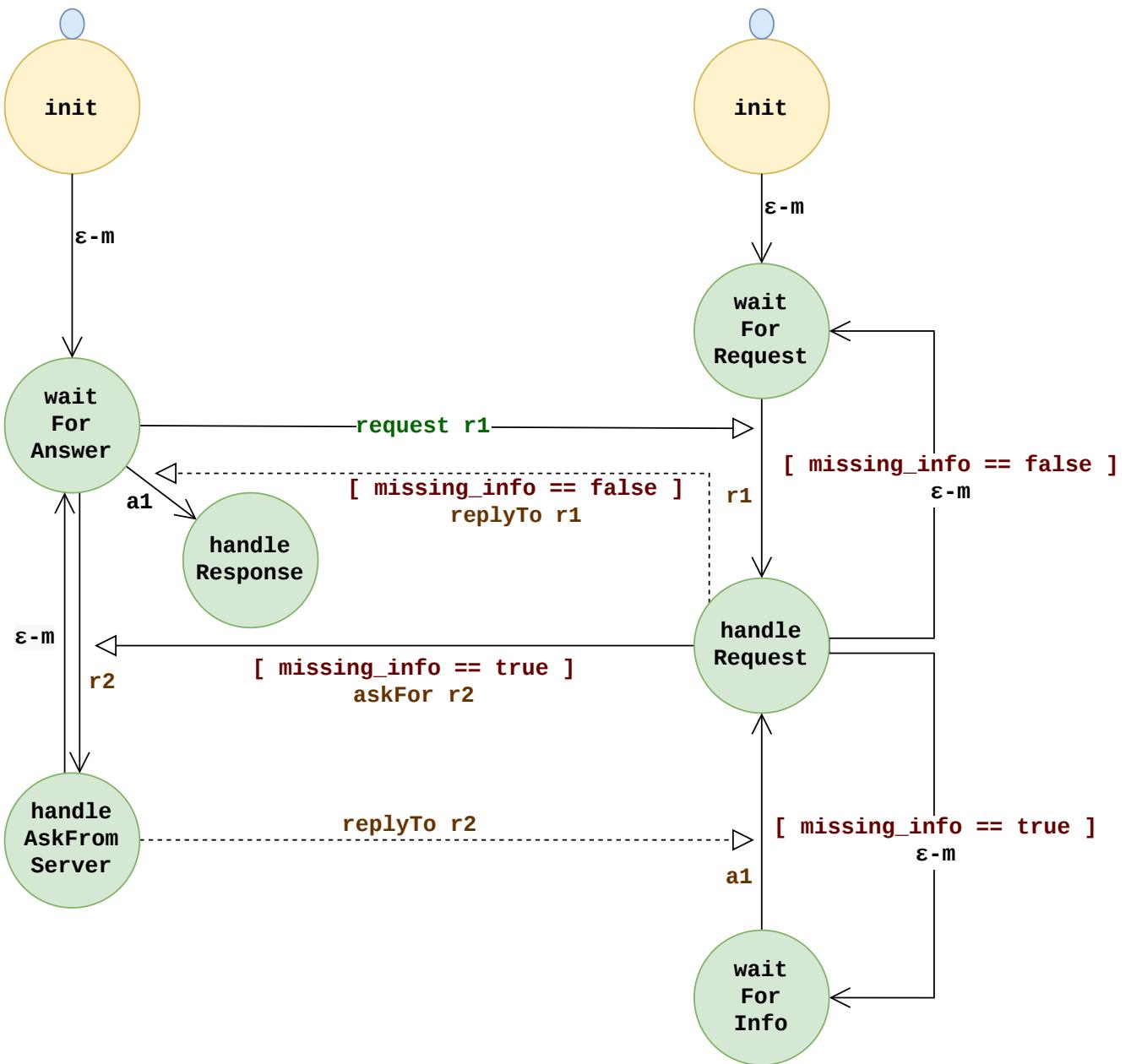
To simplify, **clientactor** is local but the same **QA-System** easily lets to communicate over the network.



## Problem analysis

[es3.qak](#)

Zooming into the actors, problem analysis produces this graph:



Notice that the state **handleRequest** of the **serviceactor** must check all received messages (both **r1** and **a1**). As you can see into the graph, if some informations are missing, then the variable **missing\_information** is set to **true** by the same state and it is used as **guard** for going to the next state. This mechanism let the service to ask for additional information more than one time.

