

# ECM3408 Enterprise Computing

## Continuous Assessment 2019-20

David Wakeling



Handed Out	Handed In
Wed 12th February 2020 (T2:04)	Mon 16th March 2020 (T2:10)

This Continuous Assessment is worth 30% of the module mark. It is an **individual assessment**, to be submitted using both E-Submit and BART. Your attention is drawn to the College and University guidelines on collaboration and plagiarism.

## 1 Specification

The e-mail standard was first proposed in 1973 at the Defense Advanced Research Projects Agency (DARPA) and later finalised by the Advanced Research Projects Agency Network (ARPANET) in 1977. This Continuous Assessment considers the microservices that might make up a simplified e-mail delivery system.

An *e-mail message* consists of three parts: the *from field* gives the source e-mail address of the message (for example, “`wilma@here.com`”); the *to field* gives the destination e-mail address of the message (for example, “`betty@there.com`”); and the *body* gives the message content.

An e-mail message is delivered over a network of e-mail servers. See Figure 1. When a user sends an e-mail message, it is uploaded to an e-mail server, where a *Mail Submission Agent* (*MSA*) puts it into the user’s **Outbox**. At any time, a user may ask the MSA to list the messages in their **Outbox**. At regular intervals, an e-mail server *Message Transfer Agent* (*MTA*) uses the MSA to read and delete a message from a user’s **Outbox**, and sends this message across the network to another e-mail server, whose MTA uses its MSA to add the message to another user’s **Inbox**. At any time, a user may ask the MSA to list the messages in their **Inbox**, or to read and delete messages from their **Inbox**. The network address of an e-mail server may be obtained by supplying the source or destination address of an e-mail message to a *Blue Book Server*.

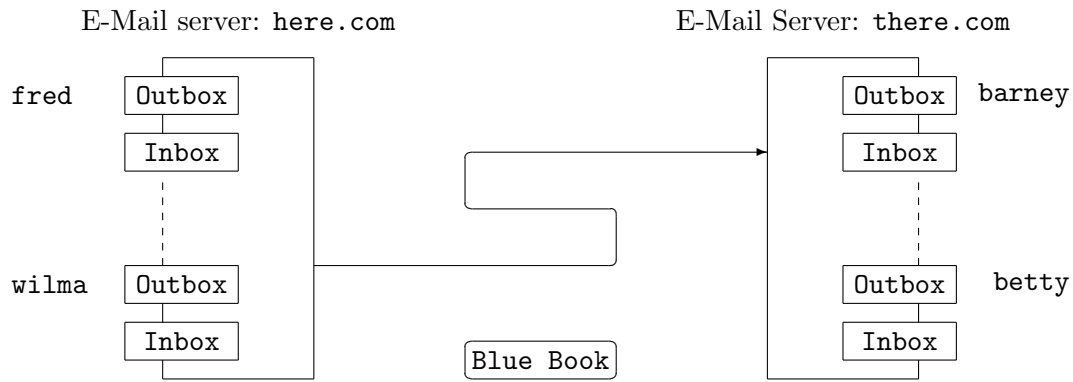


Figure 1: E-Mail servers.

Note that this description makes no reference to an e-mail *User Agent (UA)*, graphical or otherwise. Such a UA is not required — use `curl` instead.

## 2 Assessment

The assessment is organised into three parts.

### 2.1 Part 1: Microservice Identification

Briefly describe a number of microservices suitable for realising the e-mail delivery service. The description should take the form of a graph whose nodes (circles) are microservices, and whose arcs (lines) are communication channels between them. The nodes should be accompanied by a short description of what the microservice does, and the arcs should be accompanied by a short description of the data flow.

Your microservices will be judged on their appropriateness, coherence and lack of coupling. Here, appropriateness means relevance to some part of the e-mail delivery service; coherence means focus on just one part of the e-mail delivery service, and lack of coupling means independence from other parts of the e-mail delivery service.

(10 % of marks)

### 2.2 Part 2: Microservice Implementation

For each microservice that you have identified for realising the e-mail service, show some Go code that implements it. The implementation should provide a Representational State Transfer (REST) interface over the Hypertext Transfer Protocol (HTTP), and work with JavaScript Object Notation (JSON). As a simplification, none of the microservices need save any data permanently in the file system or a database.

Your microservice implementations will be judged on the clarity of their source code, the correctness of their operation, and the thoroughness of their error handling. Here, the clarity of source code refers to the layout of functions, statements and expressions, the choice of identifiers, and the use of comments; the correctness of operation refers to the processing of inputs to give outputs; and the thoroughness of error handling refers to the signalling of invalid input or output, or the failure of processing or communication with appropriate HTTP response codes. You should keep in mind the Single Responsibility Principle, the Law of Demeter, and the Don't Repeat Yourself edict.

**(70 % of marks)**

### **2.3 Part 3: Microservice Deployment**

For each microservice that you have identified for realising the e-mail delivery service, show a **Dockerfile** that could be used to provision a networked **docker** container providing the microservice, and a command that would start the container. Deploying these containerised microservices together should realise the e-mail service.

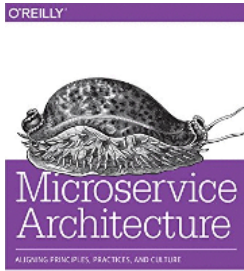
Your microservice deployment will be judged on how well the microservices work together in their networked containers.

**(20 % of marks)**

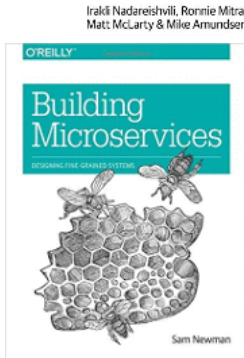
## **3 Submission**

You should submit both an electronic copy of the source code of your system of microservices and a paper copy of a listing of the same source code by midday on *Monday 16th March 2020*. The electronic copy should be submitted by E-Submit, and the paper copy by BART (including an E-Submit receipt).

## References



I. Nadareishvili, R. Mitra, M. McLarty and M. Amundsen, *Microservice Architecture: Aligning Principles, Practices and Culture*, O'Reilly, 2016, ISBN 978-1491956250



S. Newman, *Building Microservices: Designing Fine-Grained Systems*, O'Reilly, 2015, ISBN 978-1491950357.