Introduction to SMTP and e-mail

RES

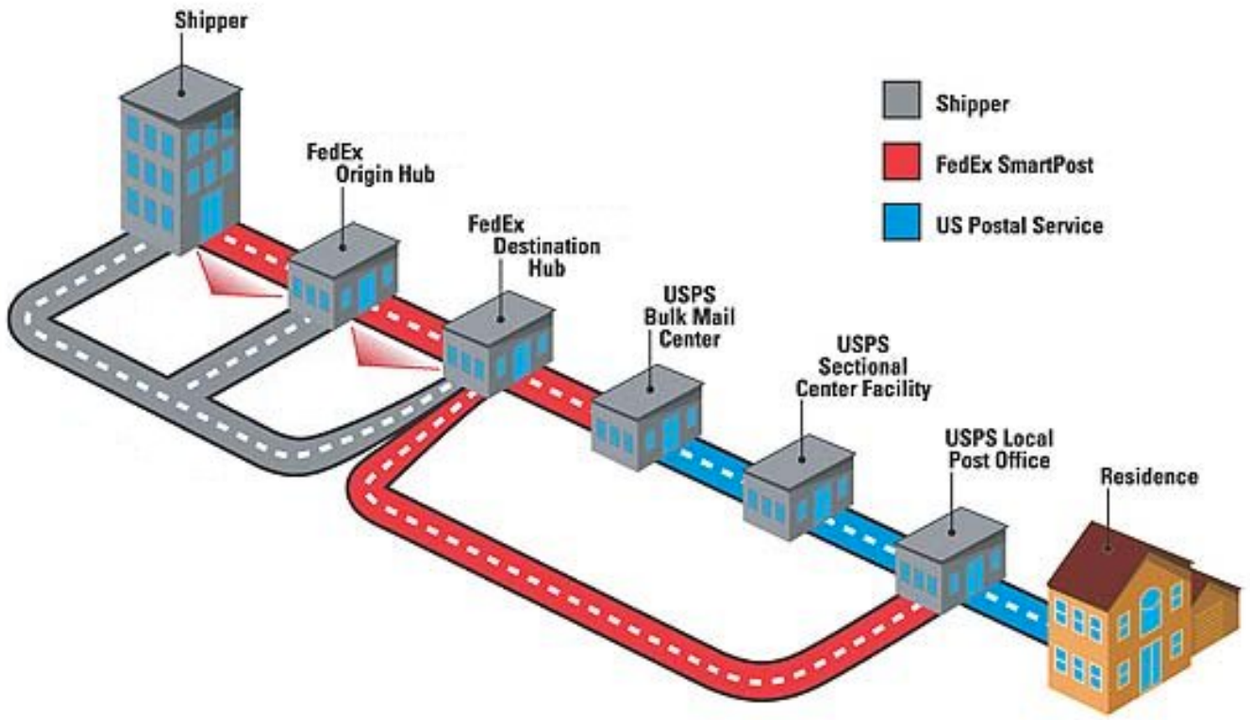
Olivier Liechti

heig-vd

Haute Ecole d'Ingénierie et de Gestion
du Canton de Vaud

heig-vd

Haute Ecole d'Ingénierie et de Gestion
du Canton de Vaud



heig-vd
Haute Ecole d'Ingénierie et de Gestion
du Canton de Vaud

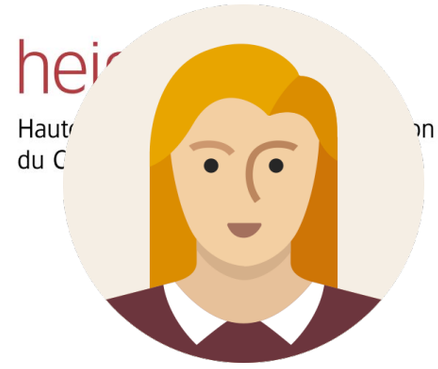
What happens when Bob wants to **send an e-mail** to Alice?

heig-vd
Haute Ecole d'Ingénierie et de Gestion
du Canton de Vaud

Bob uses **Thunderbird** to
write his mail.

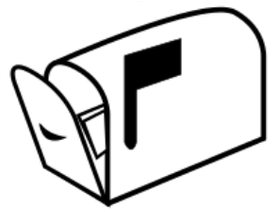
Alice uses **MS Outlook** to
check and read her mails.

In the technical specs (RFCs),
these programs are called
Mail User Agents (MUA)



Bob uses his professional e-mail address. His company runs a **MS Exchange Server**.

Alice uses her private address. She has an account (and a **mailbox**) on the **Google gmail** infrastructure.



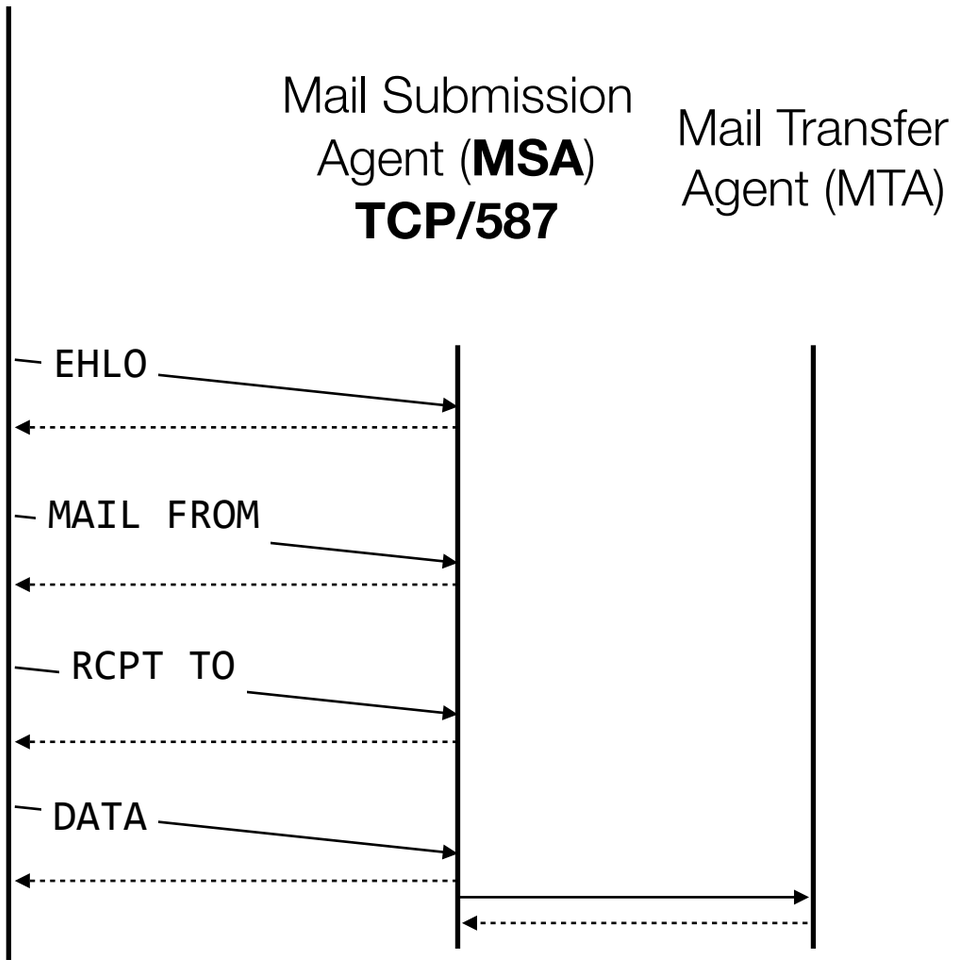
heig-vd
Haute Ecole d'Ingénierie et de Gestion
du Canton de Vaud

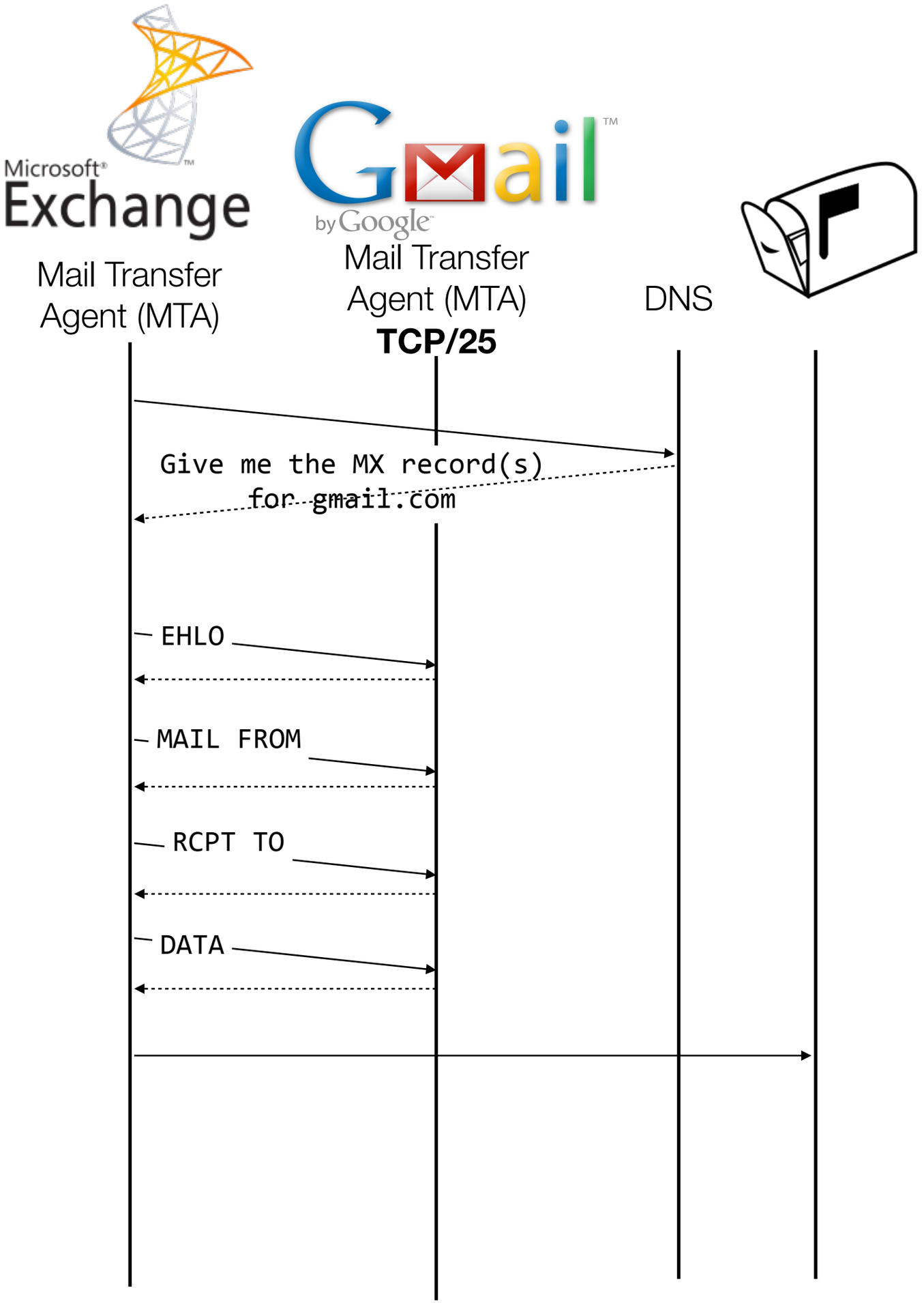
Bob writes a message to "**alice.res@gmail.com**". He pushes on the "Send" button.

The Exchange Server is made of **2 logical components:** the **MSA** and the **MTA**.

Bob's MUA asks Bob's MSA to deliver the mail. It uses the **SMTP** protocol for that purpose and (should) use TCP port 587.

After enforcing **usage policies**, the MSA delegates the work to the MTA. We don't know how.





heig-vd
Haute Ecole d'Ingénierie et de Gestion du Canton de Vaud

Bob's MTA initially does not know where to forward the mail...

It issues a **DNS** query to get a list of **MX records** for Alice's domain (gmail.com).

When Bob's MTA knows the IP address of Alice's MTA, it uses the **SMTP** protocol once more to forward the message. TCP **port 25** is used in this case.

When Alice's MTA receives the mail, it stores it in Alice's **mailbox** (for later retrieval).

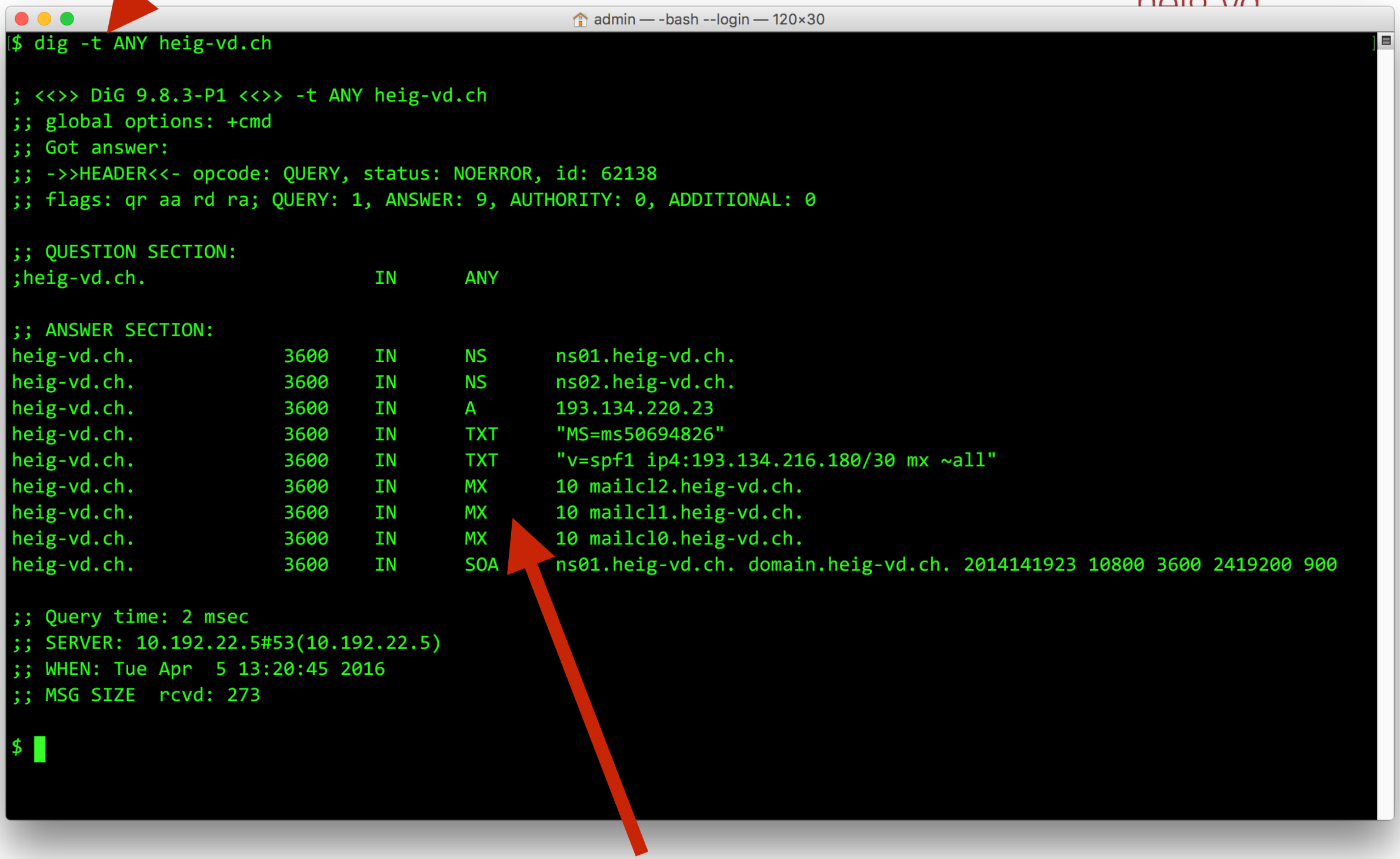
dig

admin — less ◂ man dig — 120×30

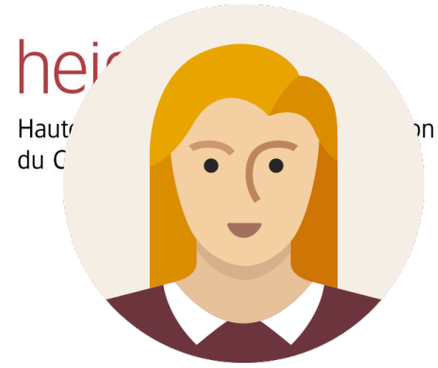
```
DIG(1)                                  BIND9                                  DIG(1)

NAME
       dig - DNS lookup utility

SYNOPSIS
       dig [@server] [-b address] [-c class] [-f filename] [-k filename] [-m] [-p port#] [-q name] [-t type]
           [-x addr] [-y [hmac:]name:key] [-4] [-6] [name] [type] [class] [queryopt...]

       dig [-h]

       dig [global-queryopt...] [query...]

DESCRIPTION
       dig (domain information groper) is a flexible tool for interrogating DNS name servers. It performs
       DNS lookups and displays the answers that are returned from the name server(s) that were queried.
       Most DNS administrators use dig to troubleshoot DNS problems because of its flexibility, ease of use
       and clarity of output. Other lookup tools tend to have less functionality than dig.

       Although dig is normally used with command-line arguments, it also has a batch mode of operation for
       reading lookup requests from a file. A brief summary of its command-line arguments and options is
       printed when the -h option is given. Unlike earlier versions, the BIND 9 implementation of dig allows
       multiple lookups to be issued from the command line.

       Unless it is told to query a specific name server, dig will try each of the servers listed in
       /etc/resolv.conf.

       When no command line arguments or options are given, dig will perform an NS query for "." (the root).

:
```

`nslookup` is another command for querying DNS

dig -t ANY heig-vd.ch



`MX records` point to the SMTP servers for the domain

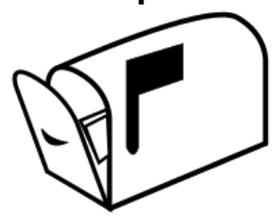


In the last step, Alice's MUA uses another protocol (e.g. IMAP, POP3) to fetch mails from the mailbox.

SMTP
587

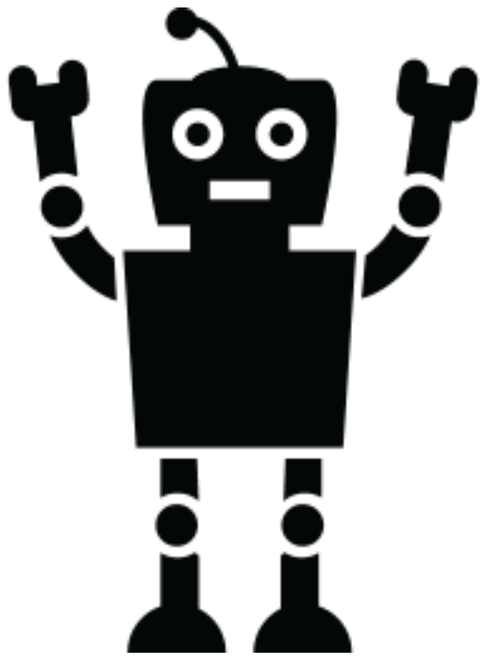
IMAP/POP3

SMTP
25



Let’s be human Exchange Servers (and play the role of Bob’s MTA).

But instead of forwarding the mail to gmail, let’s forward the mail via the **HEIG-VD’s SMTP** server.



```
dig -t MX heig-vd.ch

heig-vd.ch. 3600 INMX 10 mailcl0.heig-vd.ch.
```

```
telnet mailcl0.heig-vd.ch 25
```

```
EHLO mycompany.com
```

```
$ telnet mailcl10.heig-vd.ch 25
mailcl10.heig-vd.ch: nodename nor servname provided, or not known
$ telnet mailcl0.heig-vd.ch 25
Trying 193.134.216.181...
Connected to mailcl0.heig-vd.ch.
Escape character is '^]'.
220 heig-vd.ch ESMTP MailCleaner (Enterprise Edition 2016.01) Tue, 05 Apr 2016 14:18:24
+0200
EHLO mycompany.com
250-heig-vd.ch Hello mbp-de-admin.einet.ad.eivd.ch [10.192.116.92]
250-SIZE 20480000
250-8BITMIME
250-PIPELINING
250-AUTH PLAIN LOGIN
250-STARTTLS
250 HELP
MAIL FROM: bob@bob.com
250 OK
RCPT TO: olivier.liechti@wasabi-tech.com
250 Accepted
DATA
354 Enter message, ending with "." on a line by itself
From: bob@areyousure.com
To: olivier.liechti@wasabi-tech.com
Subject: demo

Ok. Cool. Bye.
.
250 OK id=1anPx9-0003KC-BC
quit
221 heig-vd.ch closing connection
Connection closed by foreign host.
```

SMTP command

!=

Message header

[Docs] [txt|pdf] [draft-klensin-rfc...] [Diff1] [Diff2] [Errata]

Updated by: 7504 DRAFT STANDARD
Errata Exist

Network Working Group J. Klensin
Request for Comments: 5321 October 2008
Obsoletes: 2821
Updates: 1123
Category: Standards Track

Simple Mail Transfer Protocol

Status of This Memo

This document specifies an Internet standards track protocol for the Internet community, and requests discussion and suggestions for improvements. Please refer to the current edition of the "Internet Official Protocol Standards" (STD 1) for the standardization state and status of this protocol. Distribution of this memo is unlimited.

Abstract

This document is a specification of the basic protocol for Internet electronic mail transport. It consolidates, updates, and clarifies several previous documents, making all or parts of most of them obsolete. It covers the SMTP extension mechanisms and best practices for the contemporary Internet, but does not provide details about particular extensions. Although SMTP was designed as a mail transport and delivery protocol, this specification also contains information that is important to its use as a "mail submission" protocol for "split-UA" (User Agent) mail reading systems and mobile environments.

INTERNET STANDARD
Errata Exist

Internet Engineering Task Force (IETF) R. Gellens
Request for Comments: 6409 QUALCOMM Incorporated
STD: 72 J. Klensin
Obsoletes: 4409 November 2011
Category: Standards Track
ISSN: 2070-1721

Message Submission for Mail

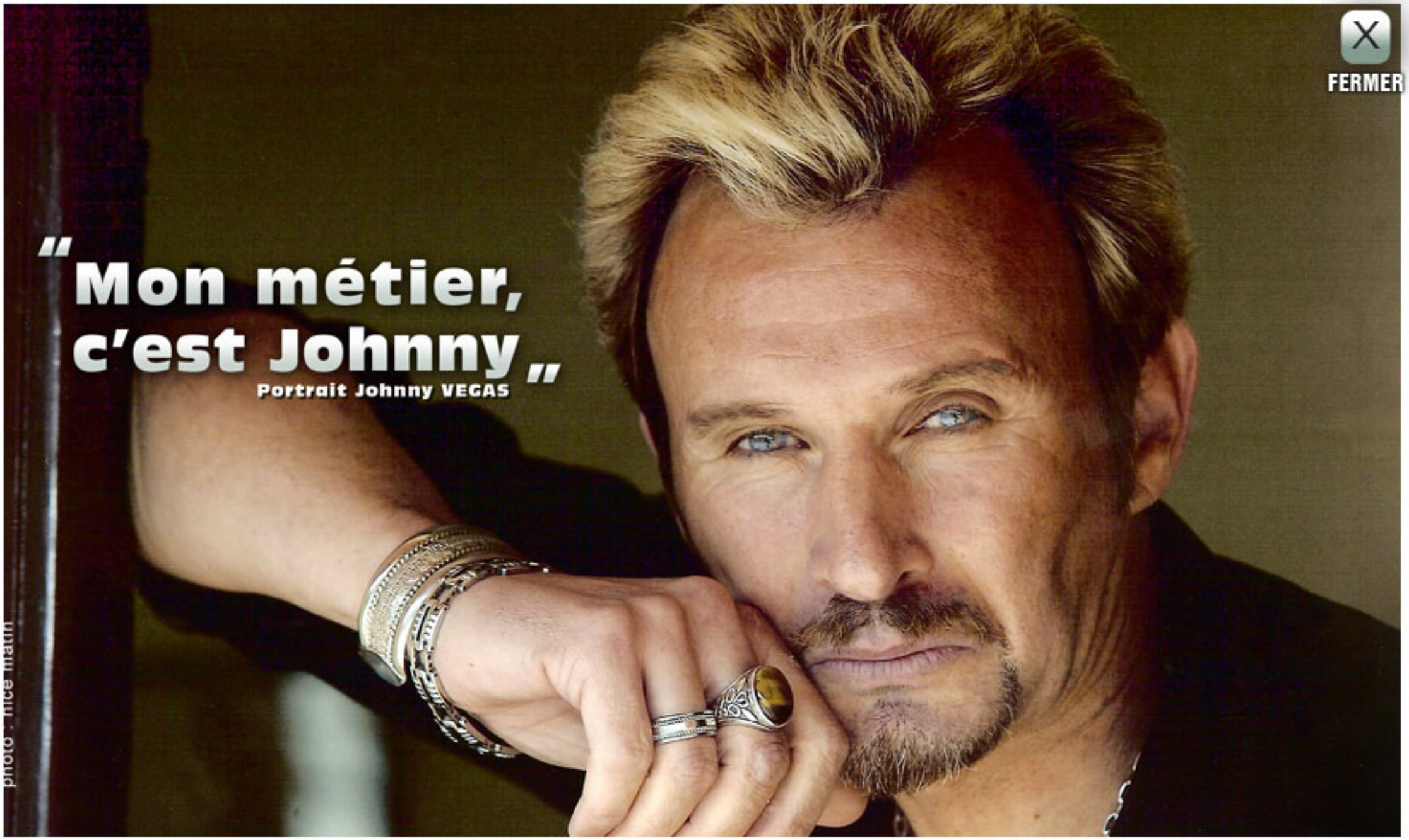
Abstract

This memo splits message submission from message relay, allowing each service to operate according to its own rules (for security, policy, etc.), and specifies what actions are to be taken by a submission server.

Message relay is unaffected, and continues to use SMTP over port 25.

When conforming to this document, message submission uses the protocol specified here, normally over port 587.

This separation of function offers a number of benefits, including the ability to apply specific security or policy requirements.



Mock Servers

Test interactions between objects

heig-vd
Haute Ecole d'Ingénierie et de Gestion
du Canton de Vaud

- **Unit tests** are meant to validate the behaviour of a single method or class.
- What should we do when we want to test a **group of classes**?
- **Example 1**
  - Think of a class "**Car**", which needs an instance of "**Engine**" to run.
  - When we test a car, do we need to build a fulle engine first?
  - How can we test the behaviour of a car when the engine overheats? How can we put the engine in this state?
- **Example 2**
  - We test a business application with a database back-end.
  - When we test the business services, will the modify the database?
  - What should we do if we wanted the database do be in a certain state before running a certain test?

TestDouble

Martin Fowler

17 January 2006

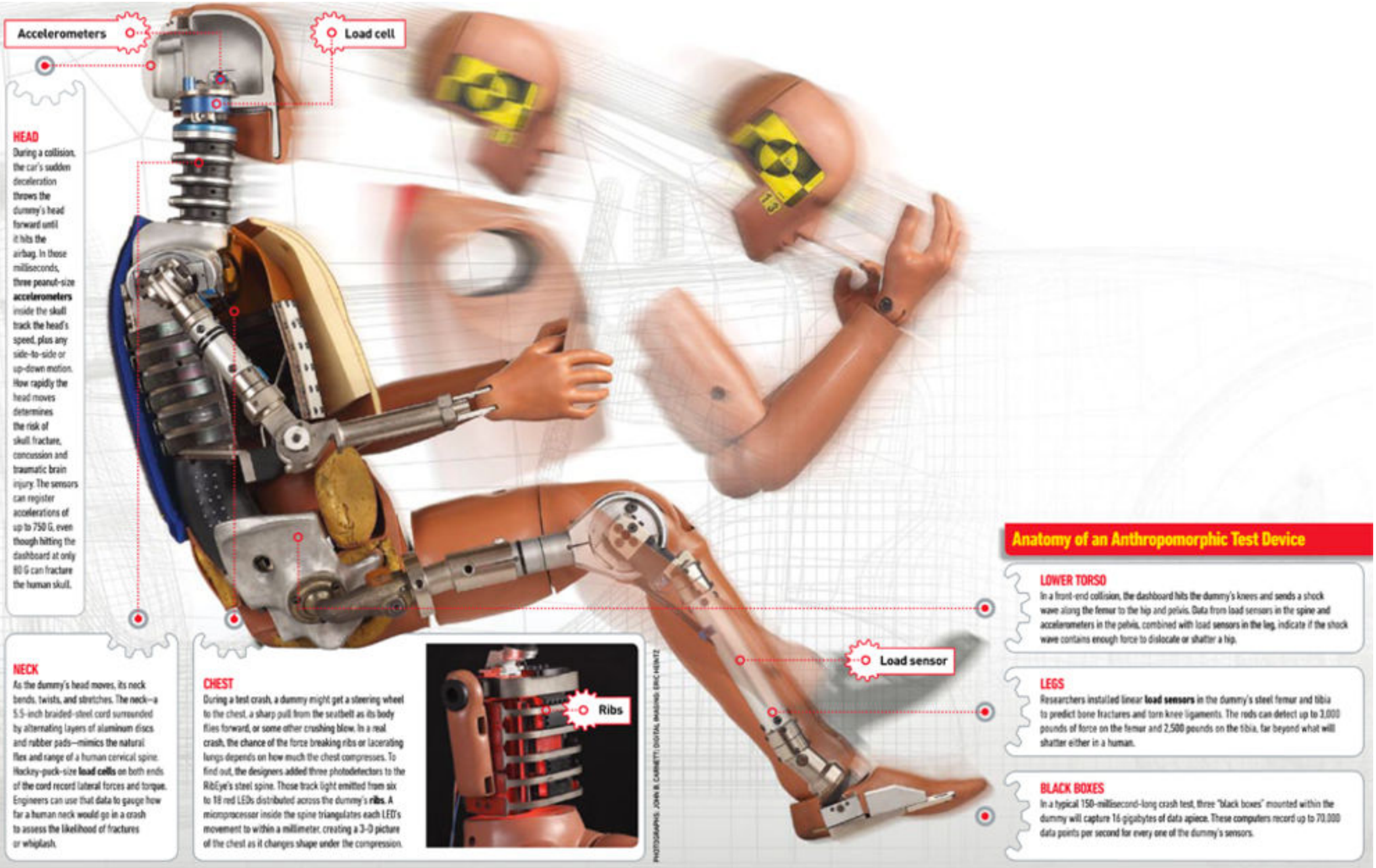
Gerard Meszaros is working on a book to capture patterns for using the various Xunit frameworks. One of the awkward things he's run into is the various names for stubs, mocks, fakes, dummies, and other things that people use to stub out parts of a system for testing. To deal with this he's come up with his own vocabulary which I think is worth spreading further.

The generic term he uses is a Test Double (think stunt double). Test Double is a generic term for any case where you replace a production object for testing purposes. There are various kinds of double that Gerard lists:

https://martinfowler.com/bliki/TestDouble.html

- **Dummy** objects are passed around but never actually used. Usually they are just used to fill parameter lists.
- **Fake** objects actually have working implementations, but usually take some shortcut which makes them not suitable for production (an InMemoryTestDatabase is a good example).
- **Stubs** provide canned answers to calls made during the test, usually not responding at all to anything outside what's programmed in for the test.
- **Spies** are stubs that also record some information based on how they were called. One form of this might be an email service that records how many messages it was sent.
- **Mocks** are pre-programmed with expectations which form a specification of the calls they are expected to receive. They can throw an exception if they receive a call they don't expect and are checked during verification to ensure they got all the calls they were expecting.

Anatomy of an Anthropomorphic Test Device



HEAD
During a collision, the car's sudden deceleration throws the dummy's head forward until it hits the airbag. In those milliseconds, three peanut-size **accelerometers** inside the skull track the head's speed, plus any side-to-side or up-down motion. How rapidly the head moves determines the risk of skull fracture, concussion and traumatic brain injury. The sensors can register accelerations of up to 750 G, even though hitting the dashboard at only 80 G can fracture the human skull.

NECK
As the dummy's head moves, its neck bends, twists, and stretches. The **neck**—a 5.5-inch braided-steel cord surrounded by alternating layers of aluminum discs and rubber pads—mimics the natural flex and range of a human cervical spine. Hockey-puck-size **load cells** on both ends of the cord record lateral forces and torque. Engineers can use that data to gauge how far a human neck would go in a crash to assess the likelihood of fractures or whiplash.

CHEST
During a test crash, a dummy might get a steering wheel to the chest, a sharp pull from the seatbelt as its body flies forward, or some other crushing blow. In a real crash, the chance of the force breaking ribs or lacerating lungs depends on how much the chest compresses. To find out, the designers added three photodetectors to the RibEye's steel spine. Those track light emitted from six to 18 red LEDs distributed across the dummy's **ribs**. A microprocessor inside the spine triangulates each LED's movement to within a millimeter, creating a 3-D picture of the chest as it changes shape under the compression.

LOWER TORSO
In a front-end collision, the dashboard hits the dummy's knees and sends a shock wave along the femur to the hip and pelvis. Data from load sensors in the spine and accelerometers in the pelvis, combined with load sensors in the leg, indicate if the shock wave contains enough force to dislocate or shatter a hip.

LEGS
Researchers installed linear **load sensors** in the dummy's steel femur and tibia to predict bone fractures and torn knee ligaments. The rods can detect up to 3,000 pounds of force on the femur and 2,500 pounds on the tibia, far beyond what will shatter either in a human.

BLACK BOXES
In a typical 150-millisecond-long crash test, three "black boxes" mounted within the dummy will capture 16 gigabytes of data apiece. These computers record up to 70,000 data points per second for every one of the dummy's sensors.

PHOTOGRAPHS: JOHN B. CARNETT/ DIGITAL IMAGING: ERIC HEINTZ

heig-vd
Haute Ecole d'Ingénierie et de Gestion
du Canton de Vaud

This repository Search Pull requests Issues Gist

tweakers-dev / MockMock

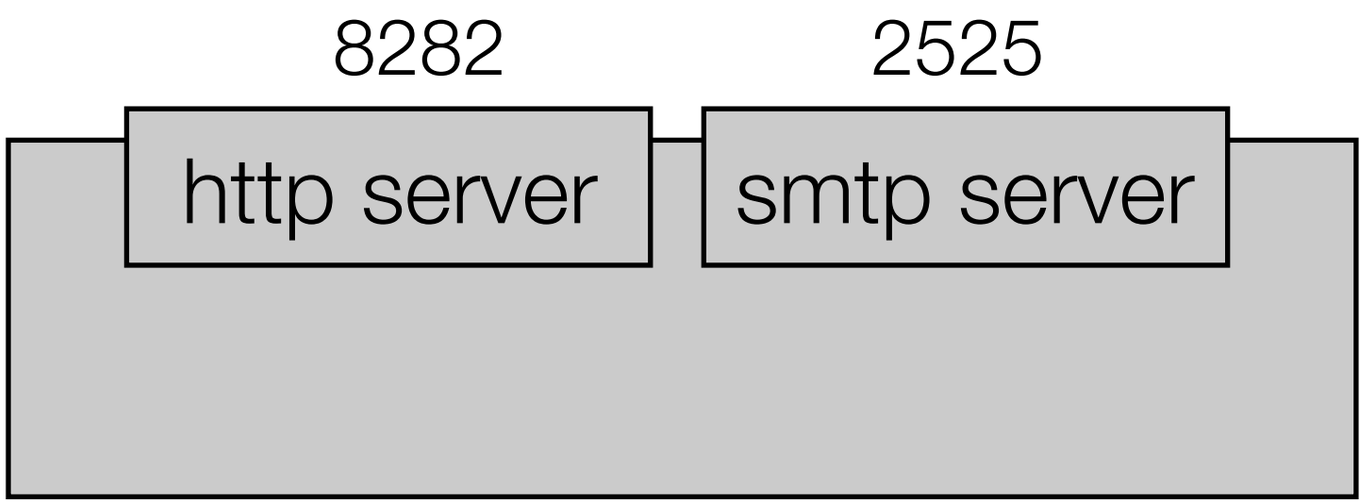
Watch 10 Star 33 Fork 15

Code Issues 1 Pull requests 2 Projects 0 Wiki Pulse Graphs

A mock SMTP server built with Java

71 commits 1 branch 0 releases 4 contributors Apache-2.0

Branch: master New pull request Create new file Upload files Find file Clone or download



MockMock Home MockMock on Github

I've got 24 mails for you. Nice! Delete all

From	To	Subject
John Doe <someone@example.org>	Some Dude <dude@examp...	Well, this is a nice subject...
John Doe <someone@example.org>	Some Dude <dude@examp...	LOL omg!
John Doe <someone@example.org>	Some Dude <dude@examp...	The iPhone 5 is huge!
John Doe <someone@example.org>	Some Dude <dude@examp...	Did you see the new MockMock version already?
John Doe <someone@example.org>	Some Dude <dude@examp...	Well, this is a nice subject...
John Doe <someone@example.org>	Some Dude <dude@examp...	Well, this is a nice subject...
John Doe <someone@example.org>	Some Dude <dude@examp...	Did you see the new MockMock version already?

heig-vd
Haute Ecole d'Ingénierie et de Gestion
du Canton de Vaud

MockMock Home MockMock on Github

Well, this is a nice subject...

Addresses

From: John Doe <someone@example.org>
To: Some Dude <dude@example.org>
CC: Barney CC Stinson <cc@example.org>

Mail headers

```
Received: from localhost (localhost [0:0:0:0:0:0:0:1])
        by 10.0.0.6
        with SMTP (MockMock SMTP Server version 1.0) id H7MBODGA;
        Thu, 27 Sep 2012 22:35:34 +0200 (CEST)
Subject: Well, this is a nice subject...
From: John Doe
To: Some Dude
Cc: Barney CC Stinson
Date: Thu, 27 Sep 2012 22:35:34 +0200
Content-Type: multipart/alternative;
 boundary="=_93eb970eb7d12e53feff4781cada58f7"
MIME-Version: 1.0
```

SMTP Labs

heig-vd
Haute Ecole d'Ingénierie et de Gestion
du Canton de Vaud

- **Objective:**
  - Build an application that generates “prank e-mails” to groups of victims and sends them via a SMTP server.
  - You don’t receive code. However, you have a series of webcasts that walk you through the design process.

- **You have different options:**
  - Watch the webcasts, study the slides and answer questions, but write no code. This has to be done by everybody and individually (up to 4.5).
  - Code, with a presentation until Thursday, April 13th (5)
  - Code, with a presentation until Monday, April 10th (5.5)
  - Code, with a presentation until Friday, April 7th (6.0)