Welcome to

0. Introduction

KEA System Integration F2021 10 ECTS

Henrik Kramselund Jereminsen hkj@zencurity.com @kramse

Slides are available as PDF, kramse@Github

0-Introduction-system-integration.tex in the repo security-courses

Course contacts

- Morten Voetmann Christiansen, morc@kea.dk
- Henrik Kramselund Jereminsen, independent network and security consultant
- Email: hkj@zencurity.dk xhek@kea.dk Mobile: +45 2026 6000

You are welcome to send emails

Plan for today

- Create a good starting point for learning
- Introduce lecturer and students
- Expectations for this course
- Literature list walkthrough

Course Materials

The materials needed for this course are in multiple parts:

- Slide shows - presentation - like this file
- Exercises - PDF which is updated along the way
- Books
- Additional resources from the internet

Note: the presentation slides are not a substitute for reading the books, papers and doing exercises, many details are not shown

I like to use Github for materials, and they are open source

Course: System Integration 10 ECTS

Teaching dates: Mondays starting 08:30

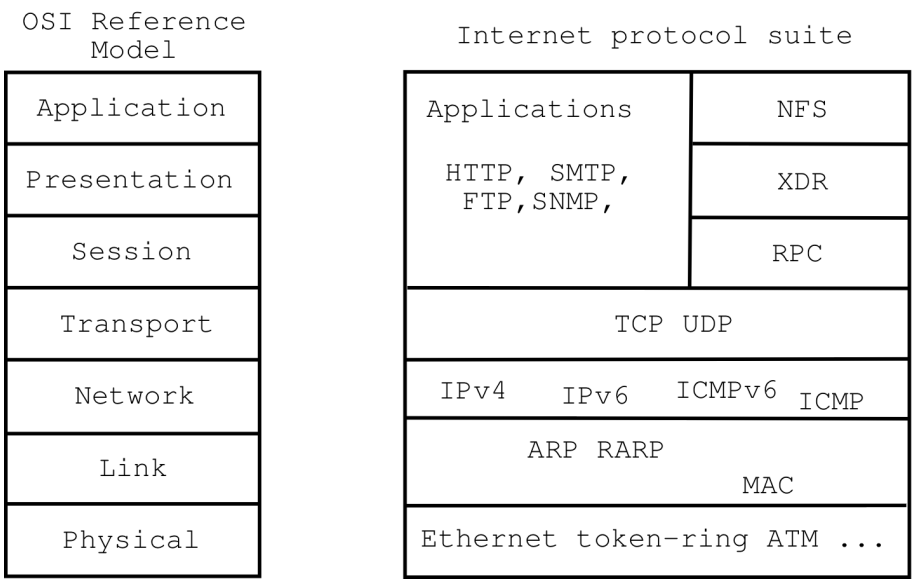
- Most weeks: 08:15 - 11:45 and 12:30 - 15:00
- Watch out for other exceptions, like easter!
- Exceptions - no teaching on these dates: March 29., April 5., May 24.

In total 14 days

Semester plan
`https://zencurity.gitbook.io/kea-it-sikkerhed/system-integration/lektionsplan`

Exam: Online exam: xxx 2020

Prerequisites



Participants are expected to have a basic understanding of software architectures and networking, as well as sufficient programming skills for independent development of software applications.

Intended Learning Outcomes

To get acquainted with the challenges of developing business applications

To understand the difference between

- tightly coupled and loosely coupled system
- synchronous and asynchronous integration

To get an overview of existing technologies and solutions in system integration

To get programming practice in developing P2P integration using networking protocols

Course Description

From: STUDIEORDNING

Knowledge

The objective is to give the student knowledge of

- business considerations associated with system integration
- standards and standardization organizations
- storage, transformation and integration of data resources
- techniques used in data conversion and migration
- the service concept and understanding of its connection with service-oriented architecture
- technologies that can be used to implement a service-oriented architecture
- integration tools

Skills

The objective is that the students acquire the ability to

- use object-oriented system in service-oriented architecture
- design a system for easy integration with other systems and using existing services
- transform or expand a system, so that it can work in a service-oriented architecture
- apply patterns that support system integration
- develop supplementary modules for generic systems
- integrate generic and other systems
- choose from different methods of integration
- translate elements of a business strategy into concrete requirements for system integration

Proficiencies

Proficiencies

The objective is that the students acquired proficiency in

- choosing from different integration techniques
- acquiring knowledge about development in standards for integration
- adapting IT architecture so that future integration of systems is taken into account
- converting elements in a business strategy to specific requirements for systems integration
- adapting a system development method, so that it supports system integration

Deliverables and Exam Procedure

- The course ends with a successful examination. The exam is individual, oral, censored, graded.
- The duration of the exam is up to 30 minutes.
- At the exam students can expect being asked any questions related to the learning objectives and presented material.

Pre-conditions
Students need to fulfill certain requirements completed mandatory tasks - to qualify for participating in the exam. Fulfilling the requirements automatically signs the student up for an exam. Alternatively, failing in delivering a mandatory task on time prevents the student from taking part in the exam.

- Deliverables:
- 2 Mandatory assignments which can be team work up to 3 students
- Both mandatory assignments are required in order to be entitled to the exam

Primary literature

Primary literature:

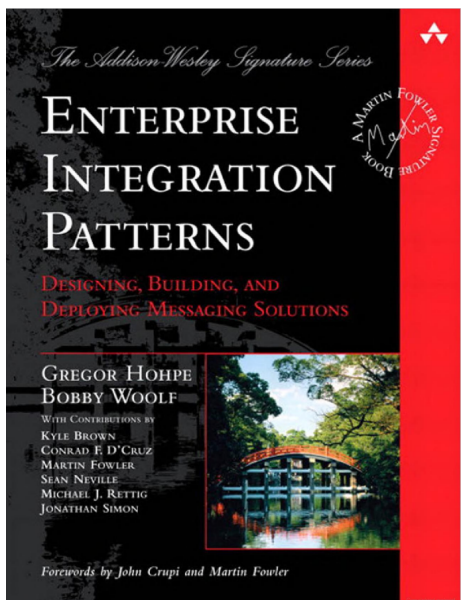
Free graphics by Lumen Design Studio

- *Enterprise Integration Patterns*, Gregor Hohpe and Bobby Woolf, 2004
  ISBN: 978-0-321-20068-6 EIP for short
- *Camel in action*, Claus Ibsen and Jonathan Anstey, 2018
  ISBN: 978-1-61729-293-4
- *Service-Oriented Architecture: Analysis and Design for Services and Microservices*,
  Thomas Erl, 2017 ISBN: 978-0-13-385858-7

Supporting literature:

- Various internet resources, to be decided

Book: Enterprise Integration Patterns



Enterprise Integration Patterns, Gregor Hohpe and Bobby Woolf, 2004
ISBN: 978-0-321-20068-6 EIP for short

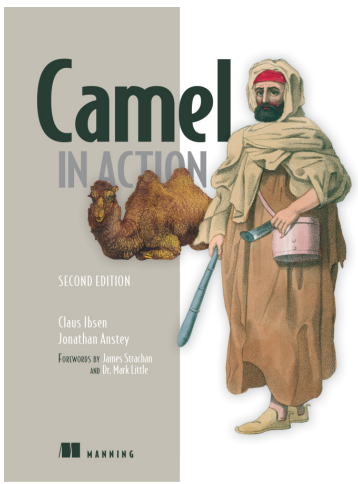
Companion Web Site

"That's why Bobby Woolf and I documented a pattern language consisting of 65 integration patterns to establish a technology-independent vocabulary and a visual notation to design and document integration solutions. Each pattern not only presents a proven solution to a recurring problem, but also documents common "gotchas"and design considerations.

The patterns are brought to life with examples implemented in messaging technologies, such as JMS, SOAP, MSMQ, .NET, and other EAI Tools. The solutions are relevant for a wide range of integration tools and platforms, such as IBM WebSphere MQ, TIBCO, Vitria, WebMethods (Software AG), or Microsoft BizTalk, messaging systems, such as JMS, WCF, Rabbit MQ, or MSMQ, ESB's such as Apache Camel, Mule, WSO2, Oracle Service Bus, Open ESB, SonicMQ, Fiorano or Fuse ServiceMix."

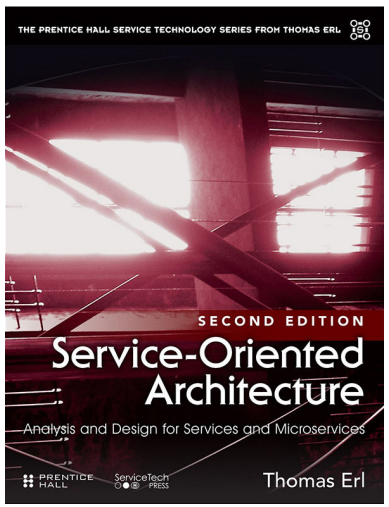
Source:
`https://www.enterpriseintegrationpatterns.com/`

Book: Camel in Action



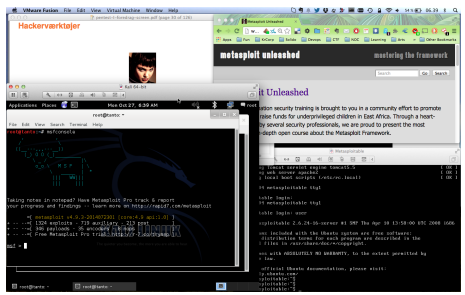
Camel in action, Claus Ibsen and Jonathan Anstey, 2018
ISBN: 978-1-61729-293-4

Book: Service-Oriented Architecture



Service-Oriented Architecture: Analysis and Design for Services and Microservices, Thomas Erl, 2017 ISBN: 978-0-13-385858-7

Lab Setup



- It will be great to have virtualisation running, to try various systems
- Hardware: modern laptop CPU with virtualisation
  Dont forget to enable hardware virtualisation in the BIOS
- Virtualisation software: VMware, Virtual box, HyperV pick your poison
- Linux server system: Debian 10 Buster amd64 64-bithttps://www.debian.org/
- Setup instructions can be found at https://github.com/kramse/kramse-labs

It is enough if these VMs are pr team

Command prompt

We will use Unix/Linux systems, and you need to use the command line a bit:

```
[hlk@fischer hlk]$ id
uid=6000(hlk) gid=20(staff) groups=20(staff),
0(wheel), 80(admin), 160(cvs)
[hlk@fischer hlk]$

[root@fischer hlk]# id
uid=0(root) gid=0(wheel) groups=0(wheel), 1(daemon),
2(kmem), 3(sys), 4(tty), 5(operator), 20(staff),
31(guest), 80(admin)
[root@fischer hlk]#
```

$ is commonly used for showing a user login, while a # is for root logins

Change from user to root using the command `sudo` like `sudo -s`

Command Syntax

A common syntax for commands are described like this:

```
echo [-n] [string ...]
```

- The command is the first thing on the command line, you cannot write `henrik echo`
- Options are prefixed with dash `-n`, optional ones are in brackets `[]`
- Multiple options can be combined into one group like, `tar -cvf` eller `tar cvf`
- Some options require arguments, like `tar -cf filename` where `-f` needs a filename

Manual System

kommando	[options]	[argumenter]
$ cal	-j	2005

It is a book about a Spanish guy called Manual. You should read it. – Dilbert

The Unix/Linux manual system is where you find the options, commands and file formats
Manuals must be installed, if not install them immediately
Very similar across Unix variants, OpenBSD is known for having an excellent manual pages
`man -k` allows keyword search similar can be done using `apropos`

Try `man crontab` and `man 5 crontab`

Example Manual Page

```
NAME
     cal - displays a calendar
SYNOPSIS
     cal [-jy] [[month]  year]
DESCRIPTION
   cal displays a simple calendar.  If arguments are not specified, the cur-
   rent month is displayed.  The options are as follows:
   -j      Display julian dates (days one-based, numbered from January 1).
   -y      Display a calendar for the current year.

The Gregorian Reformation is assumed to have occurred in 1752 on the 3rd
of September.  By this time, most countries had recognized the reforma-
tion (although a few did not recognize it until the early 1900's.)  Ten
days following that date were eliminated by the reformation, so the cal-
endar for that month is a bit unusual.
```

Unix Command Line Shells

- sh - Bourne Shell
- bash - Bourne Again Shell, often the default in Linux
- ksh - Korn shell, originally by David Korn, popular version `pdksh` public domain ksh
- csh - C shell, syntax close to the C programming language
- multiple others exist: zsh, tcsh

Comparable to command.com, cmd.exe and powershell in Windows

Also commonly used for small programs, scripts

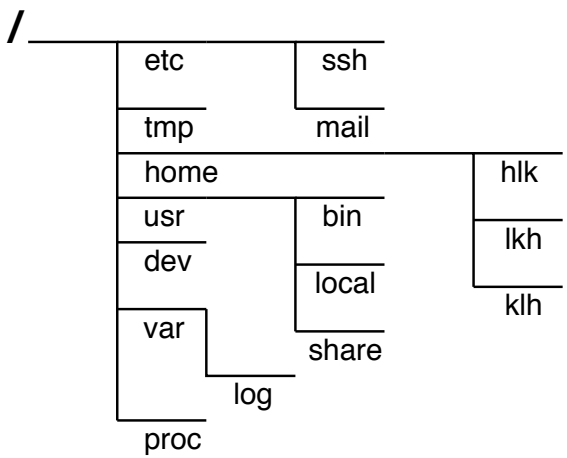
When writing scripts use the characters number sign and exclamation mark (`#!`) in the beginning

See more in `https://en.wikipedia.org/wiki/Shell_(computing)`
`https://en.wikipedia.org/wiki/Shebang_(Unix)`

Linux file system and konfiguration

.

- Unix/Linux uses a virtual filesystem
  `https://en.wikipedia.org/wiki/Unix_filesystem`
- No drive letters, just disks mounted in a common tree
- Everything starts with the file system root / - forward
- An important directory is `/etc/` which includes a lot of configuration for the system and applications



Part II

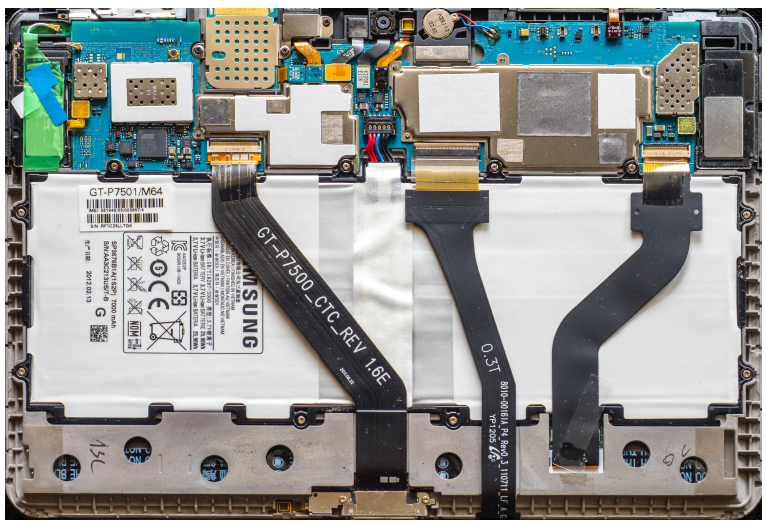
Definition

System integration is defined in engineering as the process of bringing together the component sub-systems into one system (an aggregation of subsystems cooperating so that the system is able to deliver the overarching fu that the subsystems function together as a system,[1] and in information technology[2] as the process of linking tog systems and software applications physically or functionally,[3] to act as a coordinated whole.

The system integrator integrates discrete systems utilizing a variety of techniques such as computer networking, enterprise application integration, business process management or manual programming.[4]

Source:
`https://en.wikipedia.org/wiki/System_integration`

What is Infrastructure



- Enterprises today have a lot of computing systems supporting the business needs
- These are very diverse and often discrete systems

Photo by Alexander Schimmeck on Unsplash

System Integration challenges

- Business Challenges
- Software Challenges
- Developers Challenges

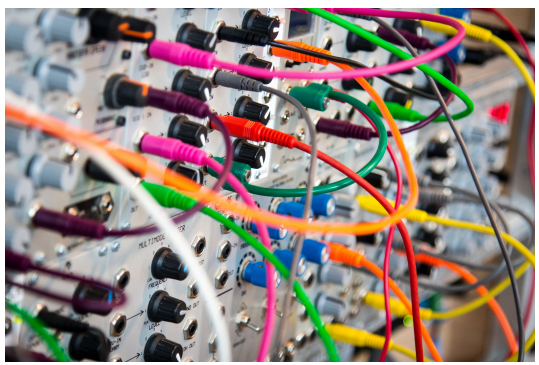
Photo by Jorik Kleen on Unsplash

Business Challenges

- Accumulation of software
- Legacy systems
- Partners
- Various types of data
- Employee churn, replacement

Photo by Adam Bignell on Unsplash

Software Challenges



- Complexity
- Various languages
- Various programming paradigms, client server, monolith, Model View Controller
- Conflicting data types and available structures
- Steam train vs electric train

Photo by John Barkiple on Unsplash

Developers Challenges

- Work in teams across organisation - and partners, vendors, sub-contractors
- Work with legacy systems, old technology
- Learn new Technologies

Photo by Kelly Sikkema on Unsplash

Integration Challenges

- Enable communication between components
- Need mediator, interpreter, translator
- Recognize standard patterns

Photo by Thomas Drouault on Unsplash

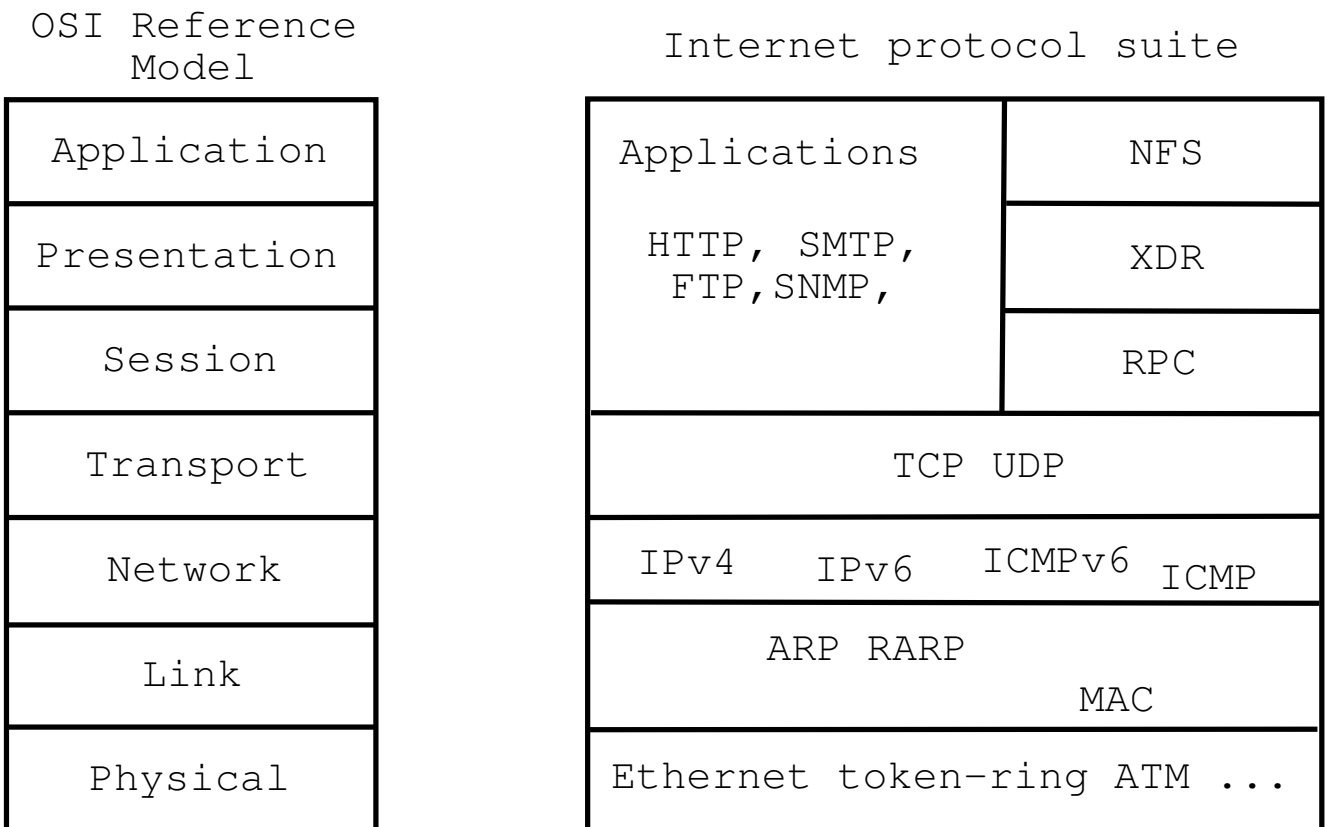
Technologies used in this course

The following tools and environments are examples that may be introduced in this course:

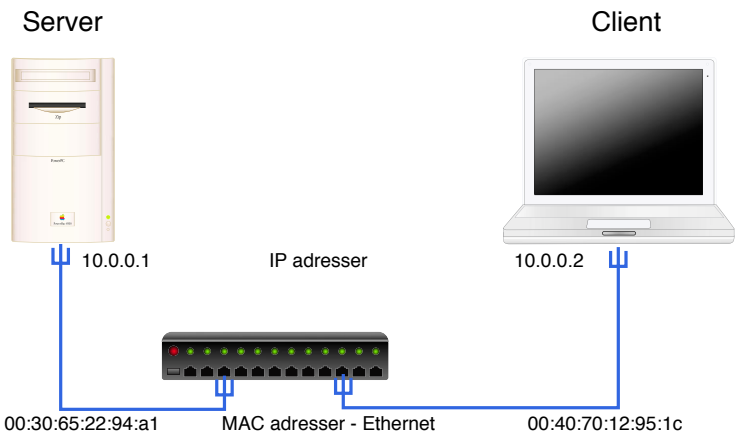
- Programming languages and frameworks Java, Spring, Python
- Development environments IDE NetBeans / Eclipse / IntelliJ, Atom
- Systems for running Java: TomCat / GlassFish
- Networking and network protocols: TCP/IP, HTTP, DNS
- Formats XML, JSON, WSDL, GRPC, msgpack, protobuf, apache thrift
- Web technologies and services: REST, API, HTML5, CSS
- Tools like cURL, Git and Github
- Integration tools Camel
- Message queueing systems: MQ
- Aggregated example platforms: Elastic stack, logstash, elasticsearch, kibana, grafana
- Cloud and virtualisation Docker, Kubernetes, Azure, AWS, microservices

This list is not complete or a promise

OSI and Internet

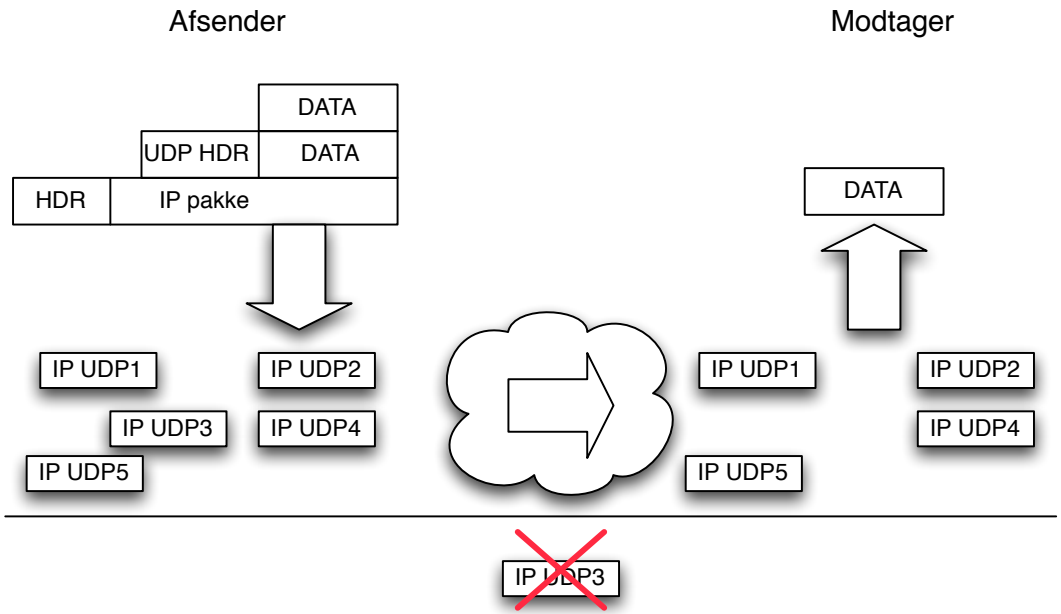


Networking in TCP/IP



- Everything uses TCP/IP today, more or less.
- Clients make requests, receives responses
- HyperText Transfer Protocol (HTTP) is an example

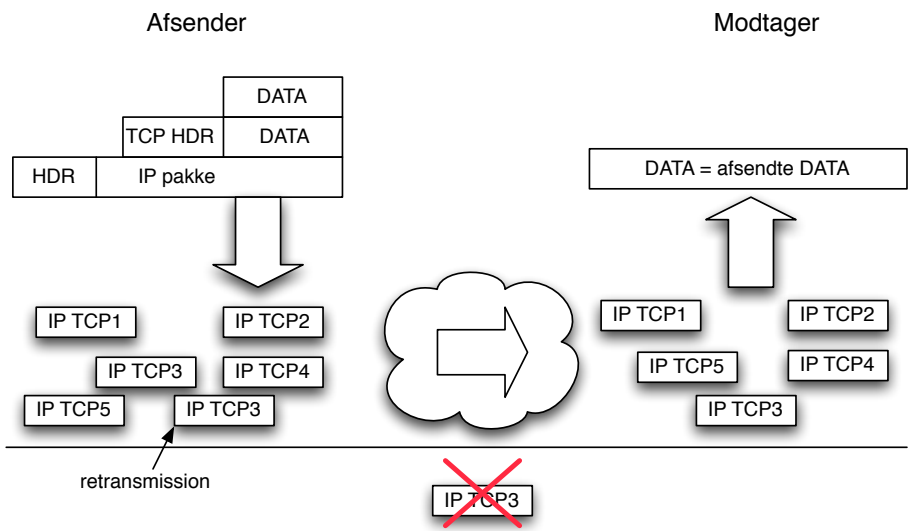
UDP User Datagram Protocol



Connection-less RFC-768, *connection-less*

Used for Domain Name Service (DNS)

TCP Transmission Control Protocol

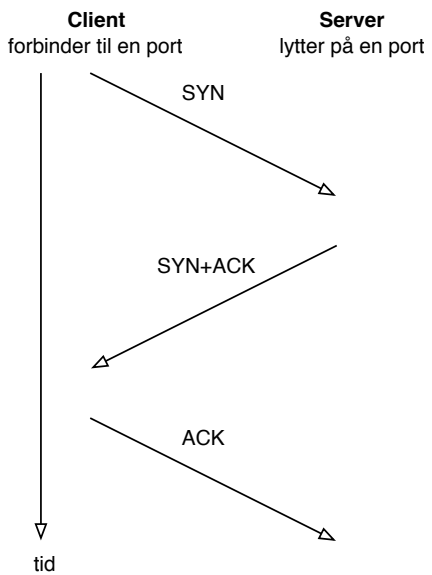


Connection oriented RFC-791 September 1981, *connection-oriented*

Either data delivered in correct order, no data missing, checksum or an error is reported

Used for HTTP and others

TCP three way handshake



- Session setup is used in some protocols
- Other protocols like HTTP/2 can perform request in the first packet

Well-known port numbers

IANA maintains a list of magical numbers in TCP/IP

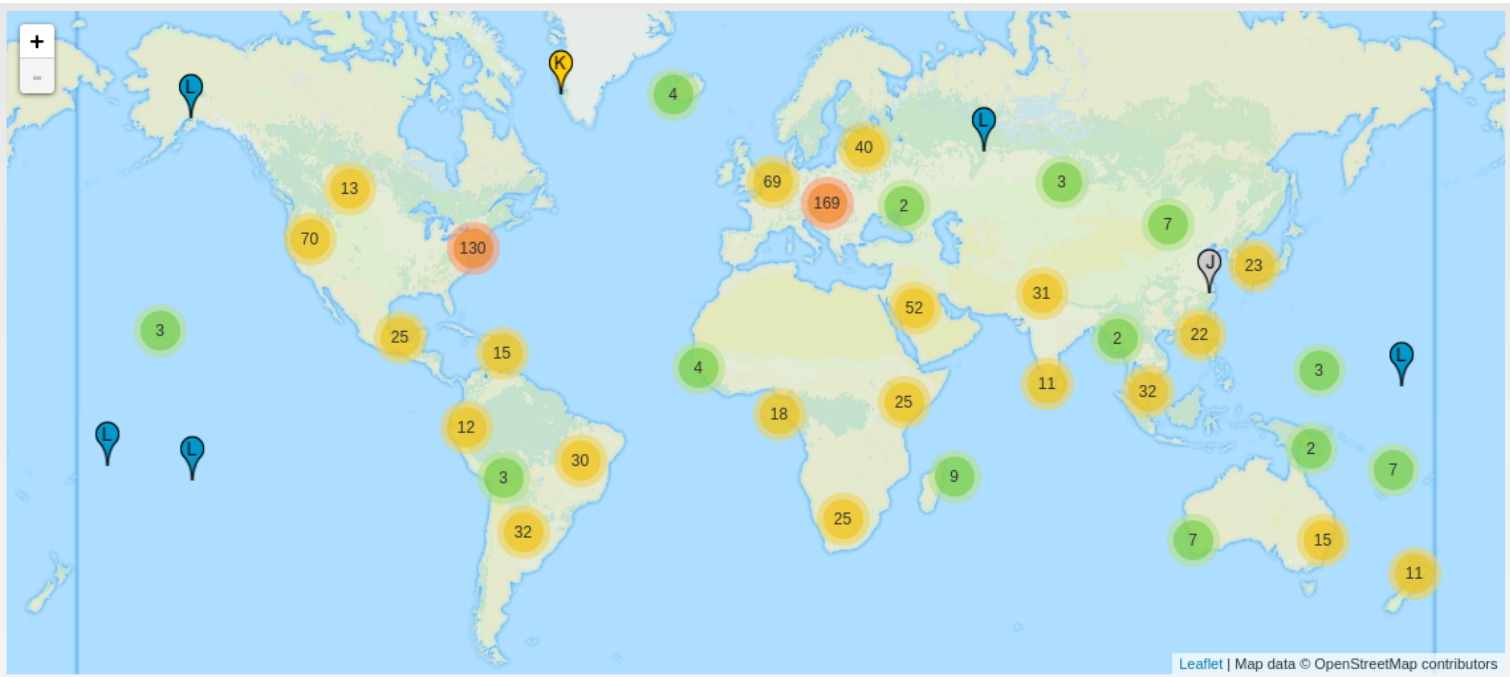
Lists of protocl numbers, port numers etc.

A few notable examples:

- Port 25/tcp Simple Mail Transfer Protocol (SMTP)
- Port 53/udp and 53/tcp Domain Name System (DNS)
- Port 80/tcp Hyper Text Transfer Protocol (HTTP)
- Port 443/tcp HTTP over TLS/SSL (HTTPS)

Source: `http://www.iana.org`

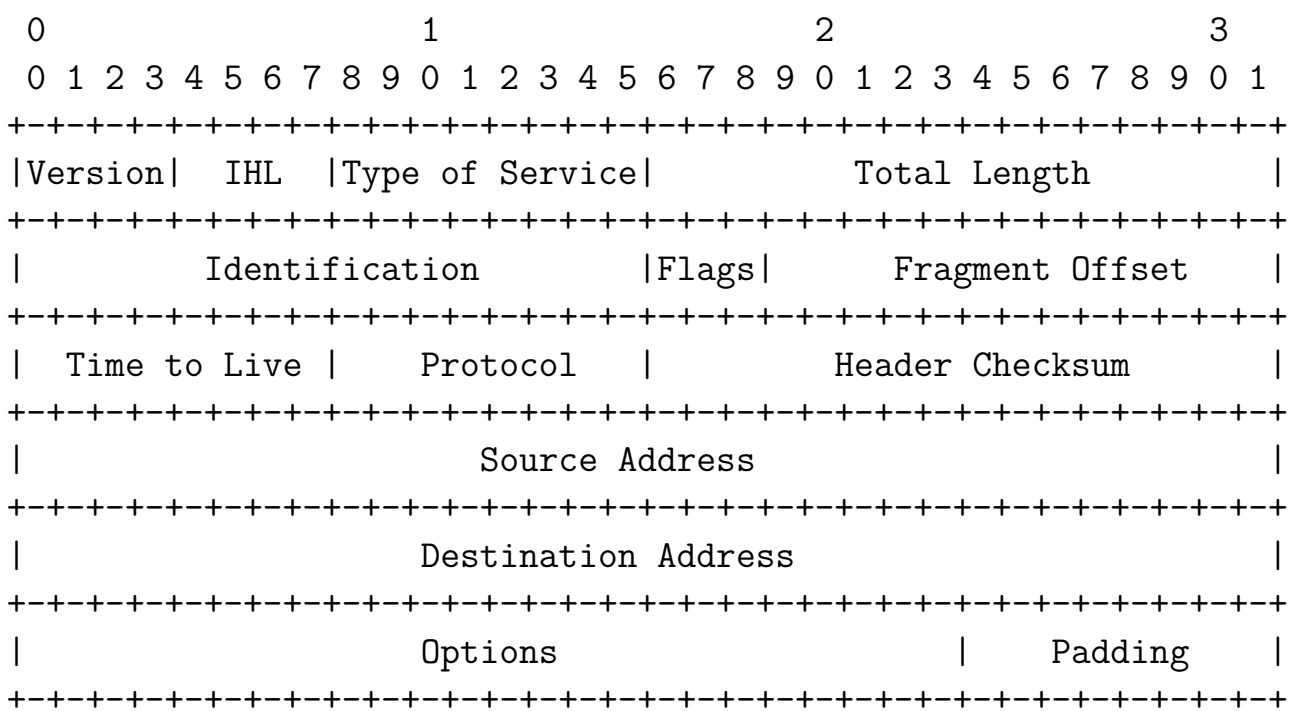
DNS root servers



As of 2019-01-29, the root server system consists of 933 instances operated by the 12 independent root server operators.

http://root-servers.org/

IPv4 packets - header - RFC-791



Example Internet Datagram Header

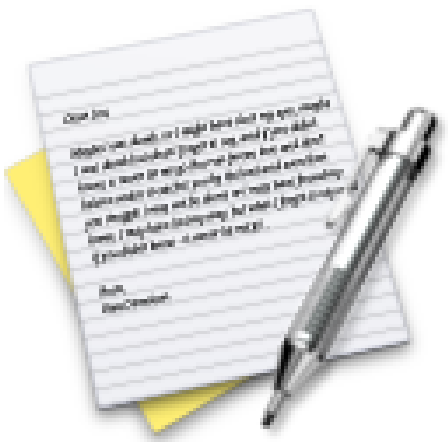
Exercise: Communicate with HTTP

Try this - use netcat/ncat, available in Nmap package from `Nmap.org`:

```
$ netcat www.zencurity.com 80
GET / HTTP/1.0

HTTP/1.1 200 OK
Server: nginx
Date: Sat, 01 Feb 2020 20:30:06 GMT
Content-Type: text/html
Content-Length: 0
Last-Modified: Thu, 04 Jan 2018 15:03:08 GMT
Connection: close
ETag: "5a4e422c-0"
Referrer-Policy: no-referrer
Accept-Ranges: bytes
...
```

Exercise

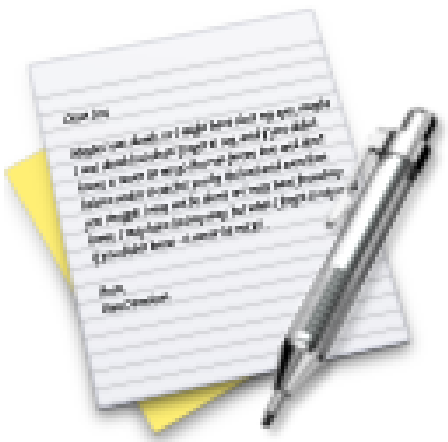


Now lets do the exercise

Getting started with the Elastic Stack - 60 min

which is number **16** in the exercise PDF.

Exercise

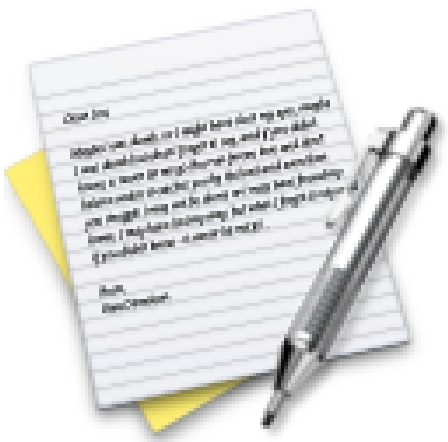


Now lets do the exercise

Grok Debugger 15 min

which is number 2 in the exercise PDF.

Exercise

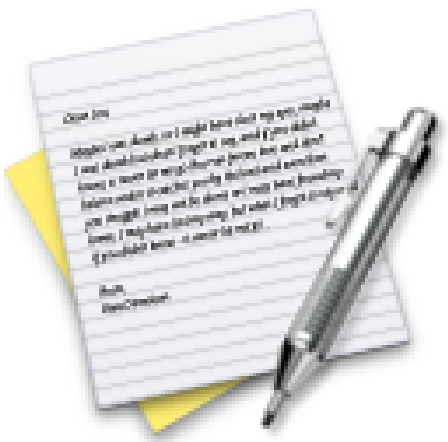


Now lets do the exercise

Getting started with the Elastic Stack 15 min

which is number **3** in the exercise PDF.

Exercise

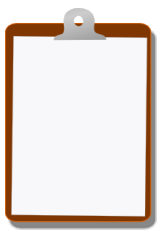


Now lets do the exercise

Check your Debian VM 10 min

which is number **4** in the exercise PDF.

For Next Time



Think about the subjects from this time, write down questions

Check the plan for chapters to read in the books

Visit web sites and download papers if needed

Retry the exercises to get more confident using the tools

Recommend buying the books!