Welcome to

1. Integration intro, Java Apps, Tomcat, XML config

KEA System Integration F2020 10 ECTS

Henrik Kramselund Jereminsen hkj@zencurity.com @kramse

Slides are available as PDF, kramse@Github

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

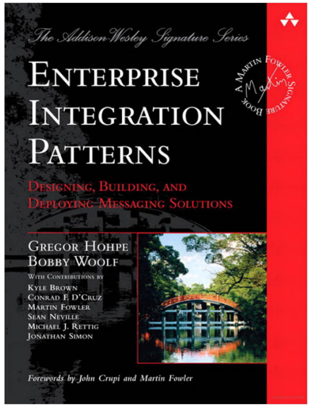
Plan for today

- Integration intro
- TCP/IP, DNS, HTTP intro
- Java Apps, Tomcat, XML config
- Git intro

Exercises

- TCP/IP
- Apache Tomcat
- Git introduction

Reading Summary



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

EIP book chapter 1-2

Definition: System integration

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 functionality) and ensuring that the subsystems function together as a system,[1] and in information technology[2] as the process of linking together different computing 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`

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/`

Integration intro

- System integration in information technology
- the process of linking together different computing systems and software applications
- Examples sales, inventory, procurement, human resources, email, web sites etc.

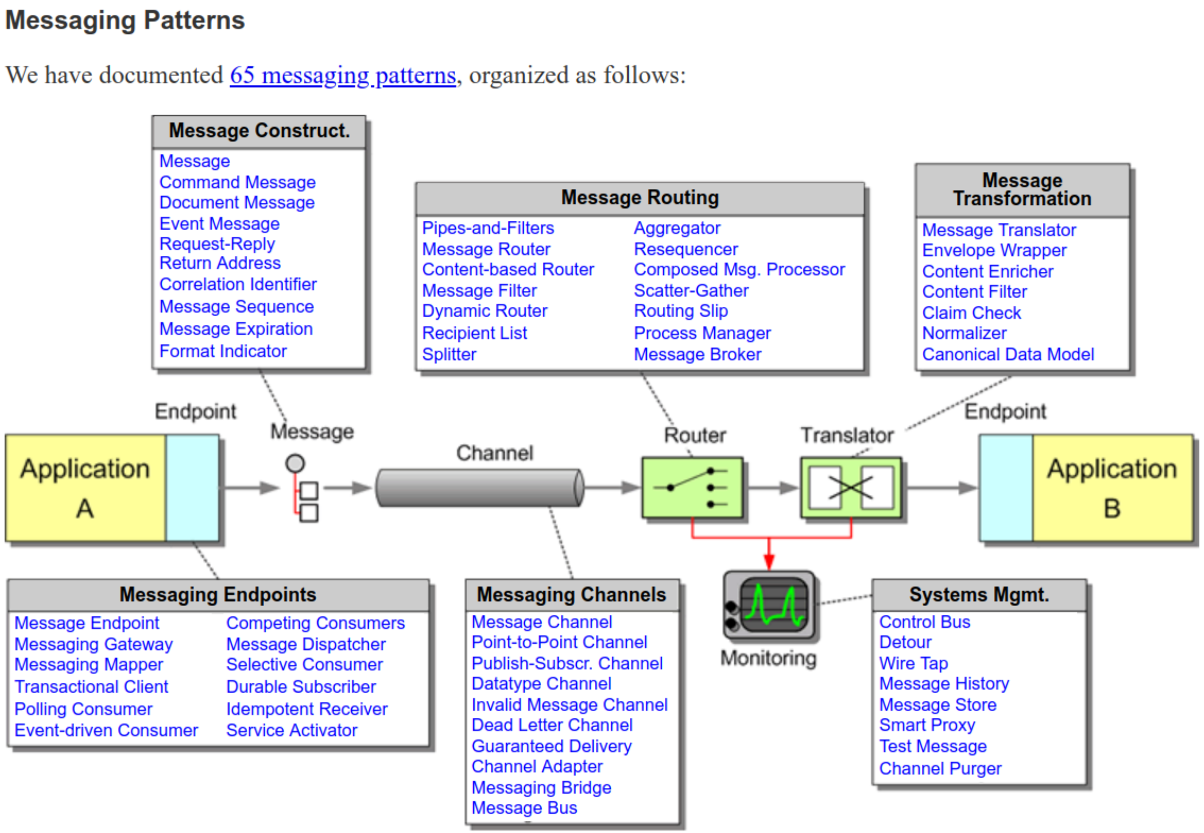
Why Enterprise Integration Patterns?

Enterprise integration is too complex to be solved with a simple ’cookbook’ approach. Instead, patterns can provide guidance by documenting the kind of experience that usually lives only in architects’ heads: they are accepted solutions to recurring problems within a given context. Patterns are abstract enough to apply to most integration technologies, but specific enough to provide hands-on guidance to designers and architects. Patterns also provide a vocabulary for developers to efficiently describe their solution.

Patterns are not ’invented’; they are harvested from repeated use in practice. If you have built integration solutions, it is likely that you have used some of these patterns, maybe in slight variations and maybe calling them by a different name. The purpose of this site is not to ”invent”new approaches, but to present a coherent collection of relevant and proven patterns, which in total form an integration pattern language.

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

EIP Patterns



Messaging Patterns

We have documented 65 messaging patterns, organized as follows:

Message Construct.

- Message
- Command Message
- Document Message
- Event Message
- Request-Reply
- Return Address
- Correlation Identifier
- Message Sequence
- Message Expiration
- Format Indicator

Message Routing

- Pipes-and-Filters
- Message Router
- Content-based Router
- Message Filter
- Dynamic Router
- Recipient List
- Splitter
- Aggregator
- Resequencer
- Composed Msg. Processor
- Scatter-Gather
- Routing Slip
- Process Manager
- Message Broker

Message Transformation

- Message Translator
- Envelope Wrapper
- Content Enricher
- Content Filter
- Claim Check
- Normalizer
- Canonical Data Model

Messaging Endpoints

- Message Endpoint
- Messaging Gateway
- Messaging Mapper
- Transactional Client
- Polling Consumer
- Event-driven Consumer
- Competing Consumers
- Message Dispatcher
- Selective Consumer
- Durable Subscriber
- Idempotent Receiver
- Service Activator

Messaging Channels

- Message Channel
- Point-to-Point Channel
- Publish-Subscr. Channel
- Datatype Channel
- Invalid Message Channel
- Dead Letter Channel
- Guaranteed Delivery
- Channel Adapter
- Messaging Bridge
- Message Bus

Systems Mgmt.

- Control Bus
- Detour
- Wire Tap
- Message History
- Message Store
- Smart Proxy
- Test Message
- Channel Purger

Challenges

- Networks are unreliable. The internet is always broken, somewhere a link is down, a system being booted etc.
- Networks are slow. Sending data across networks are slowers than making a local call
- Any two applications are different. Different programming languages, operating systems, and data formats
- Change is inevitable. Applications change over time
- Added: everything is linked, everything uses networking

Helpful patterns

- File Transfer(43)
- Shared database (47)
- Remote Procedure Invocation (50) - typically using Remote Procecure Call (RPC)
- Messaging (53) one application publishes a message to a common message channel, other applications read from the channel

Source: EIP book

Integration Styles

- Chapter 2 of the EIP book

Application coupling

Application coupling — Even integrated applications should minimize their dependencies on each other so that each can evolve without causing problems for the others. Tightly coupled applications make numerous assumptions about how the other applications work; when the applications change and break those assumptions, the integration breaks. The interface for integrating applications should be specific enough to implement useful functionality, but general enough to allow that implementation to change as needed.

Intrusivenss / Integration simplicity

Intrusivenss / Integration simplicity — When integrating an application into an enterprise, developers should strive to minimize changing the application and minimize the amount of integration code needed. Yet changes and new code will usually be necessary to provide good integration functionality, and the approaches with the least impact on the application may not provide the best integration into the enterprise.

Selecting Integration technology

Selecting Integration technology — Different integration techniques require varying amounts of specialized software and hardware. These special tools can be expensive, can lead to vendor lock-in, and increase the burden on developers to understand how to use the tools to integrate applications.

Data format

Data format — Integrated applications must agree on the format of the data they exchange, or must have an intermediate traslator to unify applications that insist on different data formats. A related issue is data format evolution and extensibility—how the format can change over time and how that will affect the applications.

Data timeliness

Data timeliness — Integration should minimize the length of time between when one application decides to share some data and other applications have that data. Data should be exchanged frequently in small chunks, rather than waiting to exchange a large set of unrelated items. Applications should be informed as soon as shared data is ready for consumption. Latency in data sharing has to be factored into the integration design; the longer sharing can take, the more opportunity for shared data to become stale, and the more complex integration becomes.

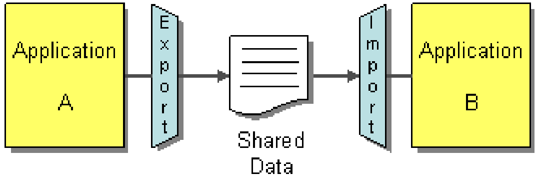
Data or functionality

Data or functionality — Integrated applications may not want to simply share data, they may wish to share functionality such that each application can invoke the functionality in the others. Invoking functionality remotely can be difficult to achieve, and even though it may seem the same as invoking local functionality, it works quite differently, with significant consequences for how well the integration works.

Remote Communication / Asynchronicity

Remote Communication / Asynchronicity — Computer processing is typically synchronous, such that a procedure waits while its subprocedure executes. It's a given that the subprocedure is available when the procedure wants to invoke it. However, a procedure may not want to wait for the subprocedure to execute; it may want to invoke the subprocedure asynchronously, starting the subprocedure but then letting it execute in the background. This is especially true of integrated applications, where the remote application may not be running or the network may be unavailable—the source application may wish to simply make shared data available or log a request for a subprocedure call, but then go on to other work confident that the remote application will act sometime later.

File Transfer

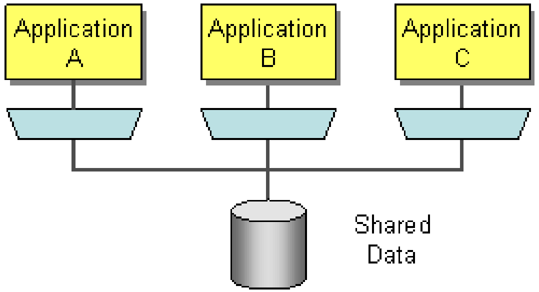


File Transfer — Have each application produce files of shared data for others to consume, and consume files that others have produced.

Common systems and technologies used:

- File Transfer Protocol (FTP) - old protocol, uses clear text password - should not be used, but still is
- SFTP/SCP - replaces FTP, Secure FTP/ Secure Copy is part of the Secure Shell (SSH) protocol - available since 1995
- Hyper Text Transfer Protocol / HTTP Secure (HTTP/HTTPS) - web based protocols

Shared Database

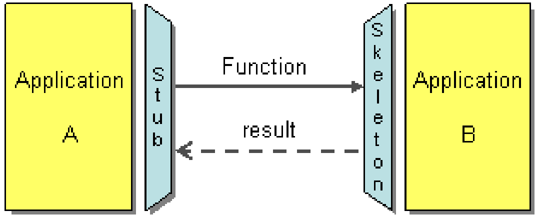


Shared Database — Have the applications store the data they wish to share in a common database.

Common systems and technologies used:

- database management system (DBMS) using Structured Query Language (SQL), relational database examples:
- PostgresSQL, Oracle DM, Microsoft SQL, MySQL `https://en.wikipedia.org/wiki/SQL`
- NoSQL databases has been a new input with examples like: MongoDB, CouchDB, Redis, RIAK `https://en.wikipedia.org/wiki/NoSQL`

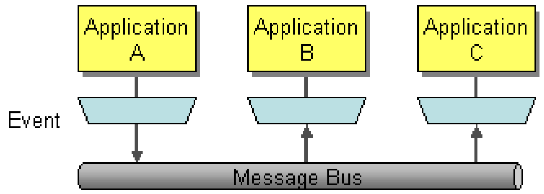
Remote Procedure Invocation



Remote Procedure Invocation — Have each application expose some of its procedures so that they can be invoked remotely, and have applications invoke those to run behavior and exchange data. Common systems and technologies used:

- Java remote method invocation (RMI), Unix RPC
- XMLHttpRequest (XHR) JavaScript in the browser makes connections and requests: `https://en.wikipedia.org/wiki/XMLHttpRequest`
- Common Object Request Broker Architecture (CORBA) used in the 1990s but not very relevant anymore
- See more at `https://en.wikipedia.org/wiki/Remote_procedure_call`

Messaging



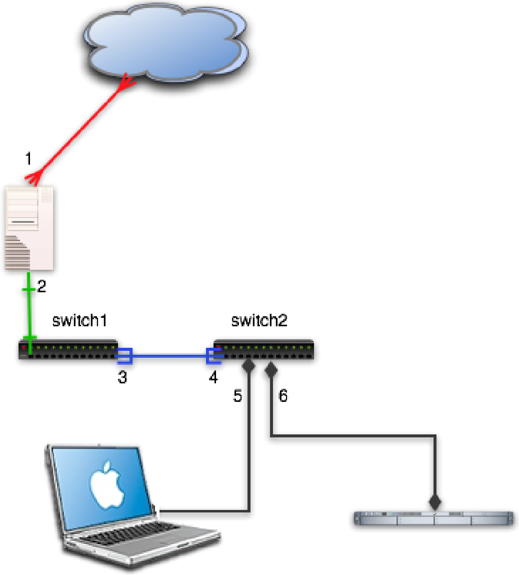
Messaging — Have each application connect to a common messaging system, and exchange data and invoke behavior using messages.

Common systems and technologies used:

- Java Message Service (JMS) API is a Java message-oriented middleware Application Programming Interface (API)
- Apache ActiveMQ, RabbitMQ, Oracle WebLogic
- See more at `https://en.wikipedia.org/wiki/Message_passing`

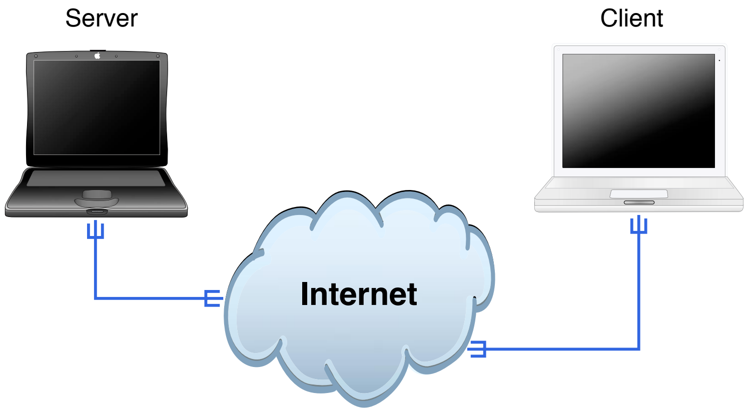
TCP/IP, DNS, HTTP intro

Networks



Our network is similar to regular networks, as found in enterprises

Internet Today



Clients and servers, roots in the academic world

Protocols are old, some more than 20 years

Very little is encrypted, mostly HTTPS

Internet is Open Standards!

We reject kings, presidents, and voting.
We believe in rough consensus and running code.
– The IETF credo Dave Clark, 1992.

Request for comments - RFC - er en serie af dokumenter

RFC, BCP, FYI, informational
de første stammer tilbage fra 1969

Ændres ikke, men får status Obsoleted når der udkommer en nyere version af en standard

Standards track:
Proposed Standard → Draft Standard → Standard

Åbne standarder = åbenhed, ikke garanti for sikkerhed

What is the Internet

Communication between humans - currently!

Based on TCP/IP

- best effort
- packet switching (IPv6 calls it packets, not datagram)
- *connection-oriented* TCP
- *connection-less* UDP

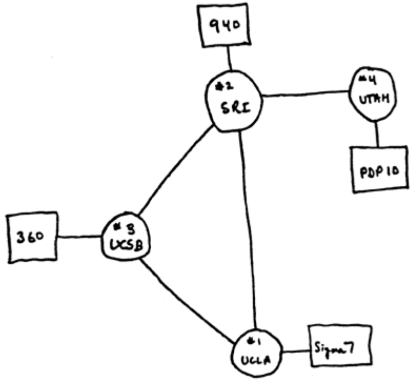
RFC-1958:

> A good analogy for the development of the Internet is that of constantly renewing the individual streets and buildings of a city, rather than razing the city and rebuilding it. The architectural principles therefore aim to provide a framework for creating cooperation and standards, as a small ”spanning set”of rules that generates a large, varied and evolving space of technology.

IP: Internet historically

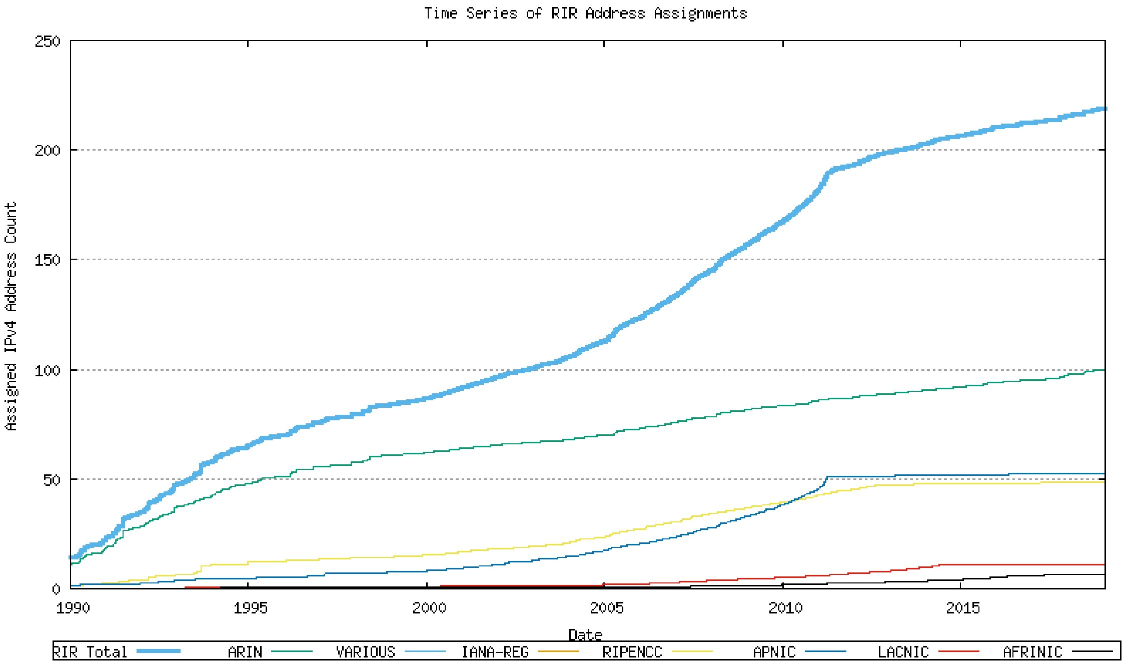
1961	L. Kleinrock, MIT packet-switching teori
1962	J. C. R. Licklider, MIT - notes
1964	Paul Baran: On Distributed Communications
1969	ARPANET start 4 nodes
1971	14 nodes
1973	Work on Internet Protocols IP started
1973	Email is about 75% af ARPANET traffik
1974	TCP/IP: Cerf/Kahn: A protocol for Packet Network Interconnection
1983	EUUG → DKUUG/DIKU connection
1988	ca. 60.000 systemer på Internet The Morris Worm rammer ca. 10%
2002	Ialt ca. 130 millioner på Internet

Internet historically set - anno 1969



- Node 1: University of California Los Angeles
- Node 2: Stanford Research Institute
- Node 3: University of California Santa Barbara
- Node 4: University of Utah

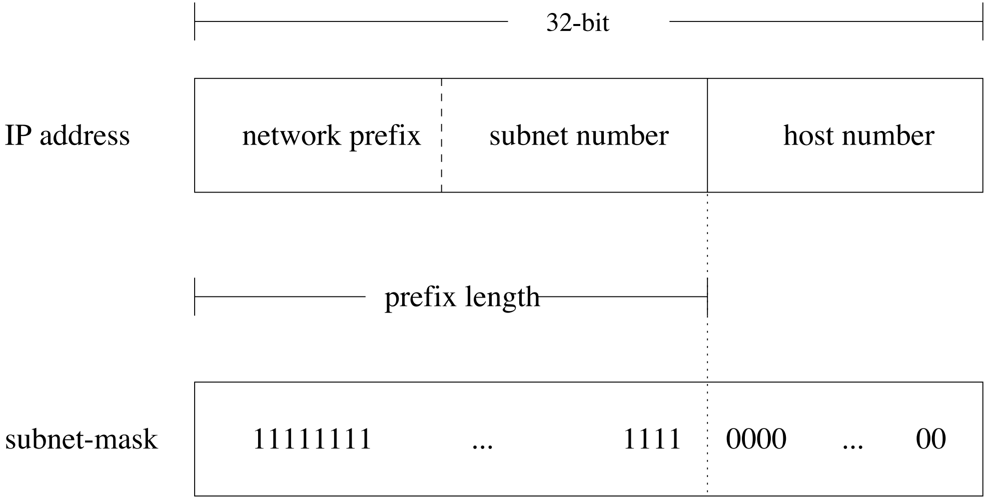
What are Internet hosts



Cumulative RIR address assignments, per RIR

Source: IPv4 Address Report - 28-Jan-2019 `http://www.potaroo.net/tools/ipv4/`

Common Address Space



Internet is defined by the address space, one

Based on 32-bit addresses, example 10.0.0.1

IPv4 address

```
hlk@bigfoot:hlk$ ipconvert.pl 127.0.0.1
Adressen er: 127.0.0.1
Adressen er: 2130706433
hlk@bigfoot:hlk$ ping 2130706433
PING 2130706433 (127.0.0.1): 56 data bytes
64 bytes from 127.0.0.1: icmp_seq=0 ttl=64 time=0.135 ms
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.144 ms
```

IP-adresser typically written as decimal numbers with dots

dot notation: 10.1.2.3

IP-adresser as bits

```
IP-adresse: 127.0.0.1
Heltal: 2130706433
Binary: 1111111000000000000000000000001
```

IP-address converted to bits

Computers use bits

Internet ABC

Previously we used classes: A, B, C, D og E

This proved to be a bit inflexible:

- A-klasse has 16 million hosts
- B-klasse about 65.000 hosts
- C-klasse only 250 hosts

Most people asked for B-klasser - starting to run out!

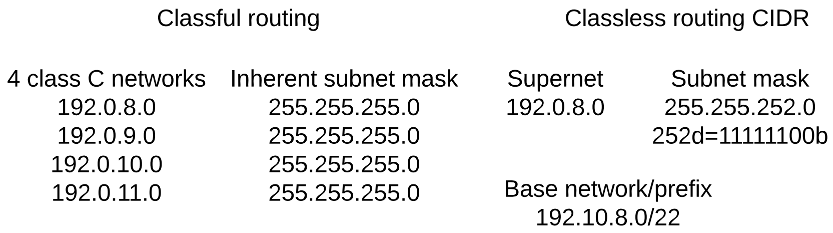
D-klasse used for multicast

E-klasse reserved

See `http://en.wikipedia.org/wiki/Classful_network`

Stop saying C, say /24

CIDR Classless Inter-Domain Routing



Classful routing

4 class C networks	Inherent subnet mask
192.0.8.0	255.255.255.0
192.0.9.0	255.255.255.0
192.0.10.0	255.255.255.0
192.0.11.0	255.255.255.0

Classless routing CIDR

Supernet	Subnet mask
192.0.8.0	255.255.252.0
	252d=11111100b

Base network/prefix
192.10.8.0/22

Subnet mask originally inferred by the class

Started to allocate multiple C-class networks - save remaining B-class
Resulted in routing table explosion

A subnet mask today is a row of 1-bit

10.0.0.0/24 means the network 10.0.0.0 with subnet mask 255.255.255.0

Supernet, supernetting

RFC-1918 Private Networks

Der findes et antal adresserum som alle må benytte frit:

- 10.0.0.0 - 10.255.255.255 (10/8 prefix)
- 172.16.0.0 - 172.31.255.255 (172.16/12 prefix)
- 192.168.0.0 - 192.168.255.255 (192.168/16 prefix)

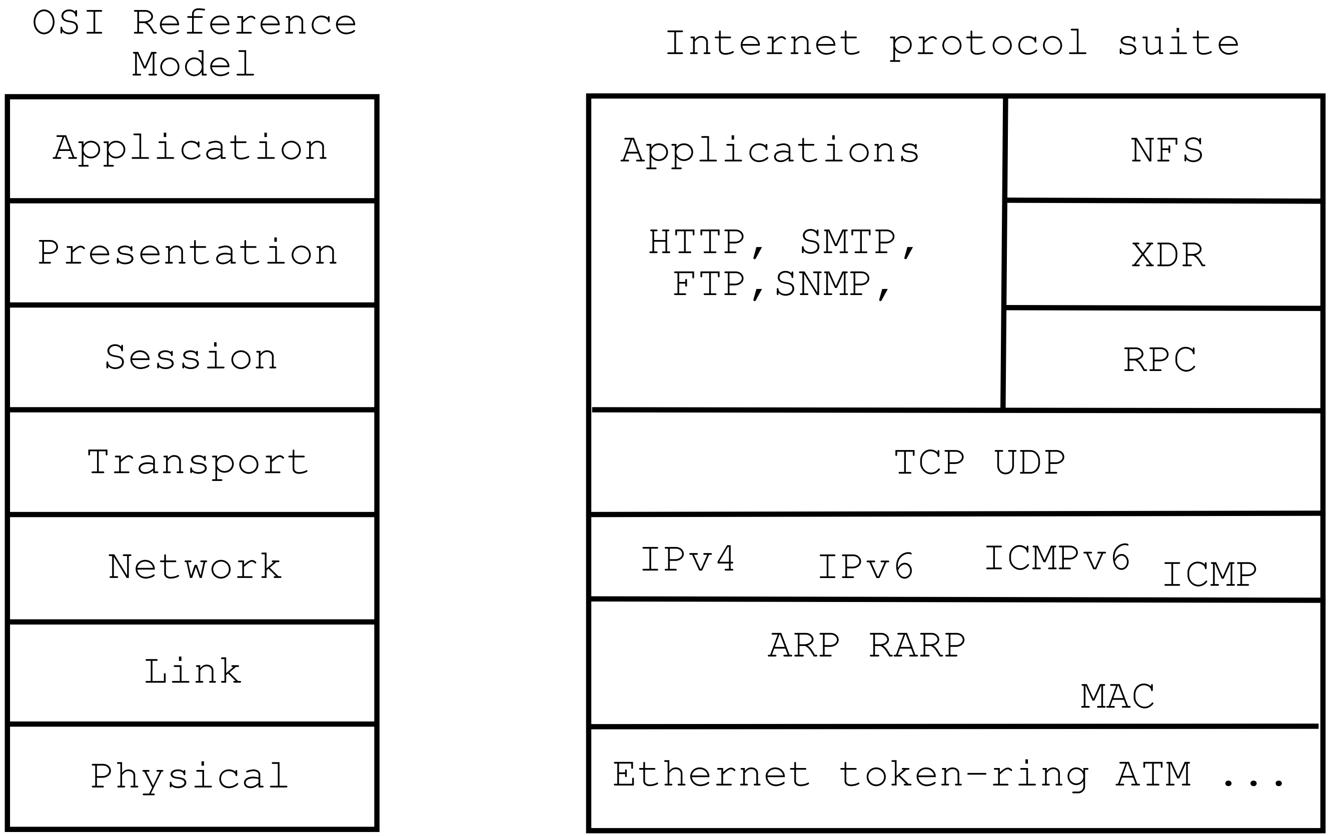
Address Allocation for Private Internets RFC-1918 adresserne!

NB: man må ikke sende pakker ud på internet med disse som afsender, giver ikke mening

```
The blocks 192.0.2.0/24 (TEST-NET-1), 198.51.100.0/24 (TEST-NET-2),
and 203.0.113.0/24 (TEST-NET-3) are provided for use in
documentation.

169.254.0.0/16 has been ear-marked as the IP range to use for end node
auto-configuration when a DHCP server may not be found
```

OSI og Internet modellerne



OSI Reference Model

Application

Presentation

Session

Transport

Network

Link

Physical

Internet protocol suite

Applications

HTTP, SMTP, FTP, SNMP,

NFS

XDR

RPC

TCP UDP

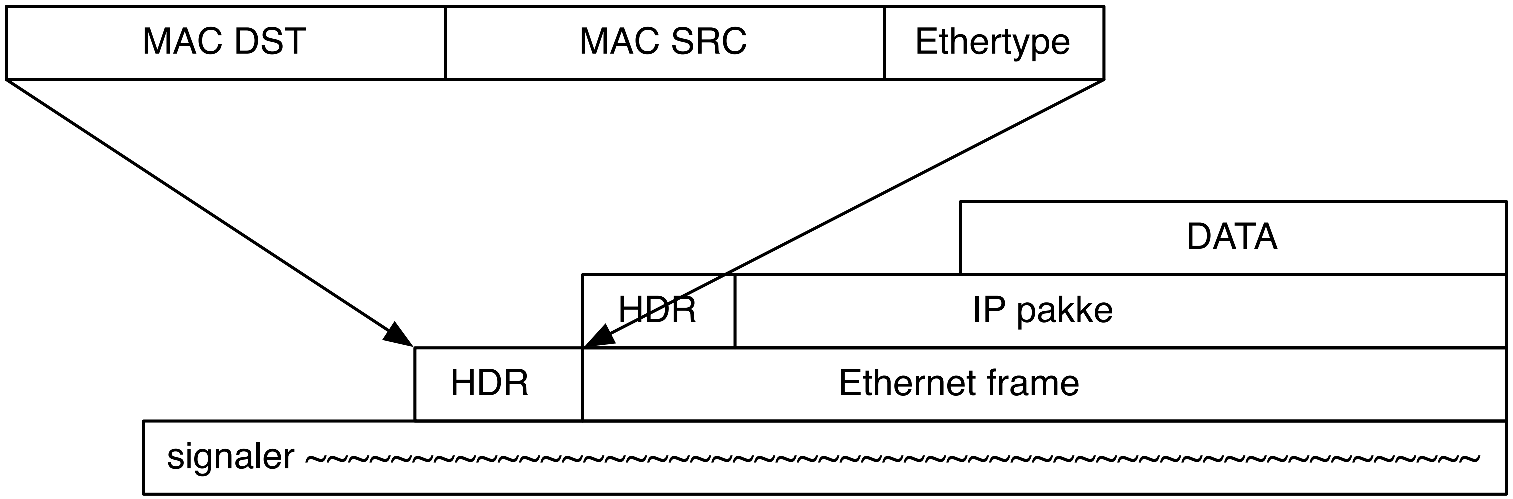
IPv4 IPv6 ICMPv6 ICMP

ARP RARP

MAC

Ethernet token-ring ATM ...

Packets across the wire or wireless


MAC DST
MAC SRC
Ethertype
DATA
HDR
IP pakke
HDR
Ethernet frame
signaler ~~~~~~

Looking at data as a stream the packets are a pattern laid on top

Network technology defines the start and end of a frame, example Ethernet

From a lower level we receive a packet, example 1500-bytes from Ethernet driver

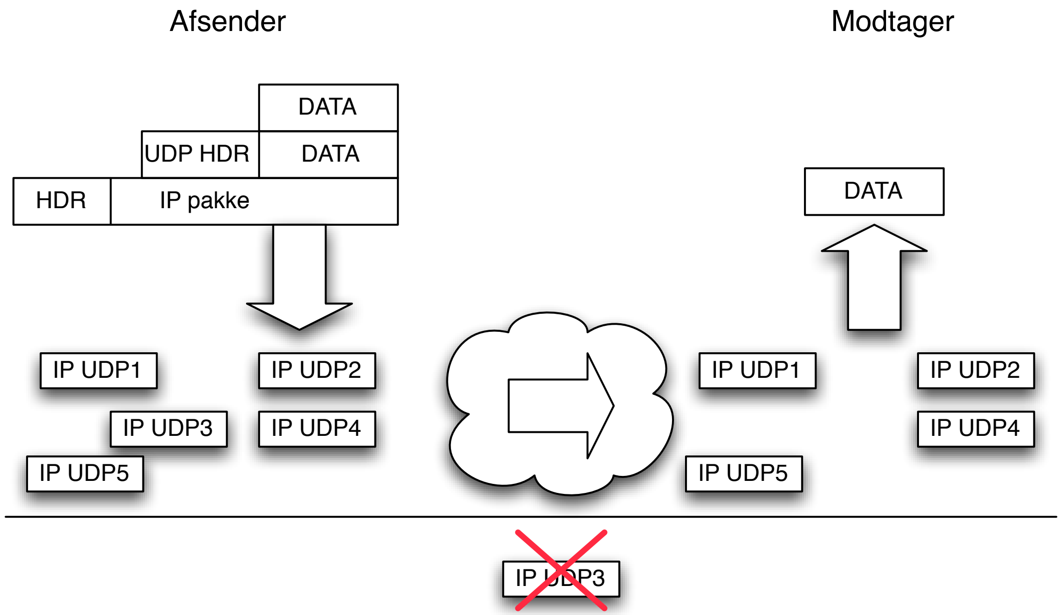
Operating system masks a lot of complexity

IPv4 pakken - header - RFC-791

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|Version|  IHL  |Type of Service|          Total Length         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         Identification        |Flags|      Fragment Offset    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Time to Live |    Protocol   |         Header Checksum       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       Source Address                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Destination Address                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Options                    |    Padding    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

                 Example Internet Datagram Header
```

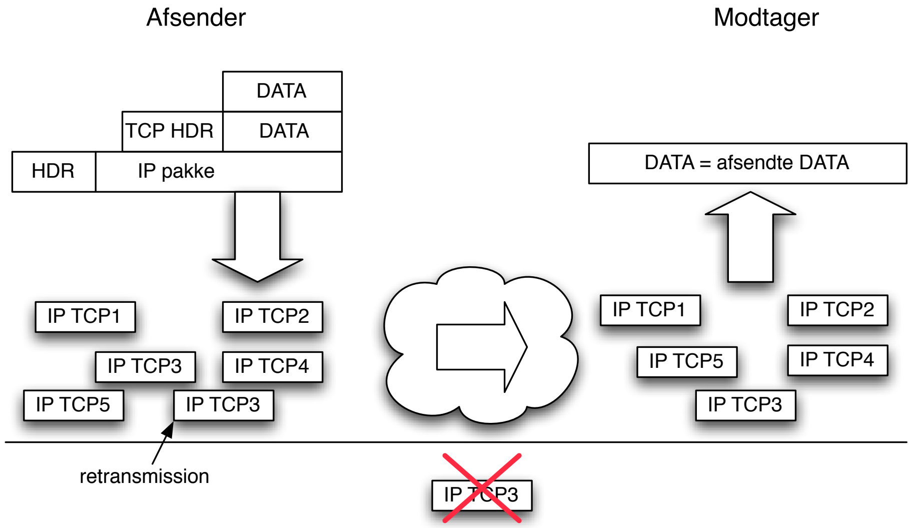
UDP User Datagram Protocol



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

Used for Domain Name Service (DNS)

TCP Transmission Control Protocol

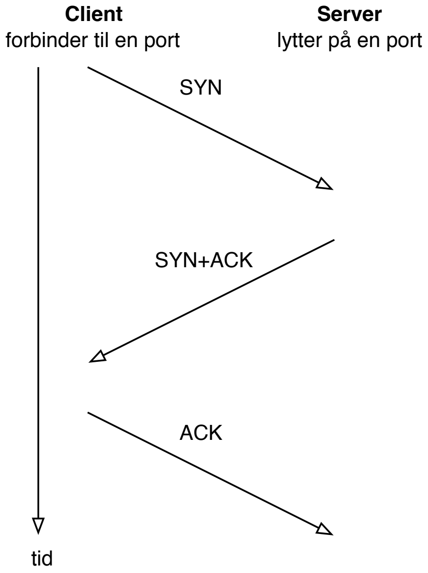


Connection oriented RFC-791 September 1981, *connection-oriented*

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`

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
...
```

Basic test tools TCP - Telnet and OpenSSL

Telnet used for terminal connections over TCP, but is clear-text

Telnet can be used for testing connections to many older protocols which uses text commands

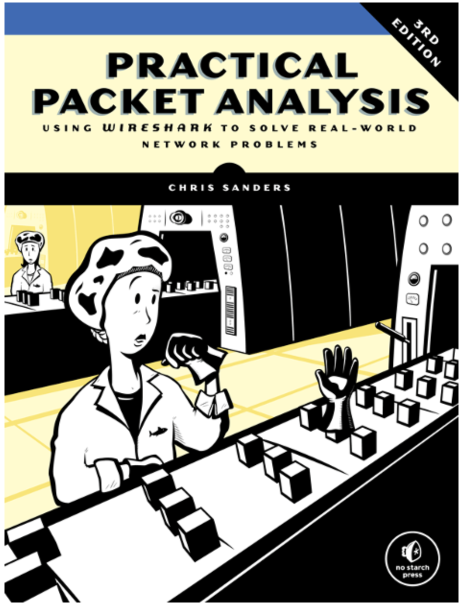
- `telnet mail.kramse.dk 25` create connection to port 25/tcp
- `telnet www.kramse.dk 80` create connection to port 80/tcp

Encrypted connections often use TLS and can be tested using OpenSSL command line tool `openssl`

- `openssl s_client -host www.kramse.dk -port 443`
  create connection to port 443/tcp with TLS
- `openssl s_client -host mail.kramse.dk -port 993`
  create connection to port 993/tcp with TLS

Using OpenSSL in client-mode allows the use of the same commands like Telnet after connection

Book: Practical Packet Analysis (PPA)


3RD EDITION
PRACTICAL PACKET ANALYSIS
USING WIRESHARK TO SOLVE REAL-WORLD NETWORK PROBLEMS
CHRIS SANDERS
no starch press

Practical Packet Analysis, Using Wireshark to Solve Real-World Network Problems by Chris Sanders, 3rd Edition April 2017, 368 pp. ISBN-13: 978-1-59327-802-1

https://nostarch.com/packetanalysis3

Apache Tomcat

> The Apache Tomcat® software is an open source implementation of the Java Servlet, JavaServer Pages, Java Expression Language and Java WebSocket technologies. The Java Servlet, JavaServer Pages, Java Expression Language and Java WebSocket specifications are developed under the Java Community Process.

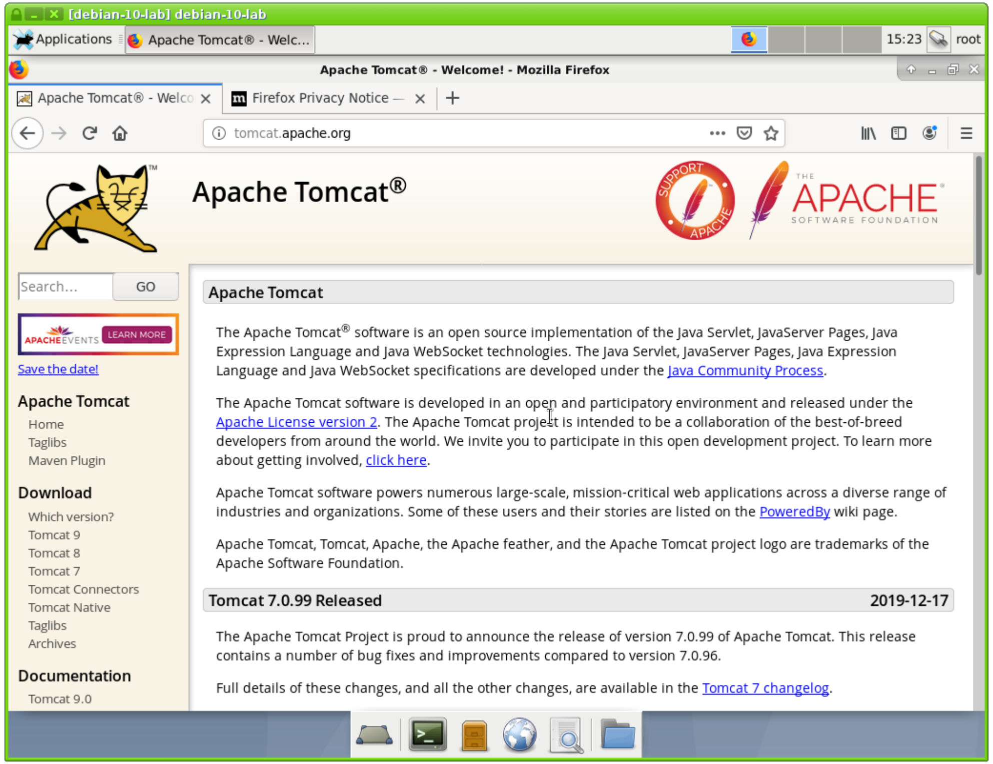
- Allows the deployment of web applications J2EE `https://en.wikipedia.org/wiki/Java_Platform,_Enterprise_Edition`
- Allows the use of Java security policies
- Contains the core functionality found in commercial packages
- `http://tomcat.apache.org/`

Java Apps, Tomcat, XML config

We will download Apache Tomcat, and perform the following:

- Download the software use version 9.0.30
  I downloaded `apache-tomcat-9.0.30.tar.gz`, Windows users take the `.zip`
- Unpack and Run the software, see it works
- Work with Tomcat manager
- Check the configuration - which is in XML
- Change the configuration - make the software listen on all IPs, specific IP

Apache Tomcat Download



Apache Tomcat®

Search... GO

Save the date!

Apache Tomcat
- Home
- Taglibs
- Maven Plugin

Download
- Which version?
- Tomcat 9
- Tomcat 8
- Tomcat 7
- Tomcat Connectors
- Tomcat Native
- Taglibs
- Archives

Documentation
- Tomcat 9.0

Apache Tomcat

The Apache Tomcat® software is an open source implementation of the Java Servlet, JavaServer Pages, Java Expression Language and Java WebSocket technologies. The Java Servlet, JavaServer Pages, Java Expression Language and Java WebSocket specifications are developed under the Java Community Process.

The Apache Tomcat software is developed in an open and participatory environment and released under the Apache License version 2. The Apache Tomcat project is intended to be a collaboration of the best-of-breed developers from around the world. We invite you to participate in this open development project. To learn more about getting involved, click here.

Apache Tomcat software powers numerous large-scale, mission-critical web applications across a diverse range of industries and organizations. Some of these users and their stories are listed on the PoweredBy wiki page.

Apache Tomcat, Tomcat, Apache, the Apache feather, and the Apache Tomcat project logo are trademarks of the Apache Software Foundation.

Tomcat 7.0.99 Released 2019-12-17

The Apache Tomcat Project is proud to announce the release of version 7.0.99 of Apache Tomcat. This release contains a number of bug fixes and improvements compared to version 7.0.96.

Full details of these changes, and all the other changes, are available in the Tomcat 7 changelog.

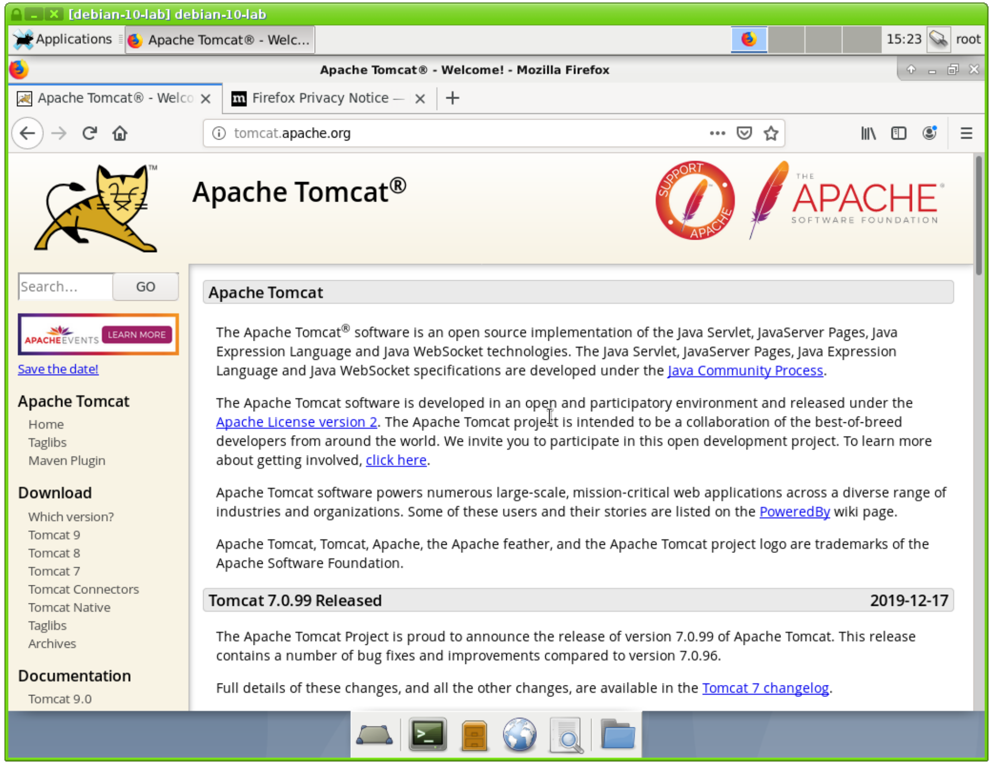
Unpack and run

Debian instructions - shortened:

- `mkdir projects;cd projects` -
- `tar zxvf $HOME/Downloads/apache-tomcat-9.0.30.tar.gz` -
- `cd apache-tomcat-9.0.30` -
- `sh bin/startup.sh` -

Windows - double click and unpack should produce the directory, and use the `startup.bat`

Tomcat Running



Apache Tomcat®

Search... GO

Save the date!

Apache Tomcat

- Home
- Taglibs
- Maven Plugin

Download

- Which version?
- Tomcat 9
- Tomcat 8
- Tomcat 7
- Tomcat Connectors
- Tomcat Native
- Taglibs
- Archives

Documentation

- Tomcat 9.0

Apache Tomcat

The Apache Tomcat® software is an open source implementation of the Java Servlet, JavaServer Pages, Java Expression Language and Java WebSocket technologies. The Java Servlet, JavaServer Pages, Java Expression Language and Java WebSocket specifications are developed under the Java Community Process.

The Apache Tomcat software is developed in an open and participatory environment and released under the Apache License version 2. The Apache Tomcat project is intended to be a collaboration of the best-of-breed developers from around the world. We invite you to participate in this open development project. To learn more about getting involved, click here.

Apache Tomcat software powers numerous large-scale, mission-critical web applications across a diverse range of industries and organizations. Some of these users and their stories are listed on the PoweredBy wiki page.

Apache Tomcat, Tomcat, Apache, the Apache feather, and the Apache Tomcat project logo are trademarks of the Apache Software Foundation.

Tomcat 7.0.99 Released 2019-12-17

The Apache Tomcat Project is proud to announce the release of version 7.0.99 of Apache Tomcat. This release contains a number of bug fixes and improvements compared to version 7.0.96.

Full details of these changes, and all the other changes, are available in the Tomcat 7 changelog.

Now use a little time to look into the server, links, which ports are open - 8080/tcp and 8009/tcp

Tomcat Tasks

- Try going to `http://127.0.0.1:8080/manager/` - get 401 Unauthorized
  Read and fix the problem
- Look into the XML file `conf/server.xml`
  Change the port 8080 into something else, does port 80/tcp work?
- XML configuration files are very common in the Java and system integration space
- Tools often found in Java world, Ant, Maven, Tomcat, Spring
- Read `https://tomcat.apache.org/tomcat-9.0-doc/monitoring.html` about Java Management Extensions (JMX) `https://en.wikipedia.org/wiki/Java_Management_Extensions`
  Note: references to RMI, JMS, SNMP, HTTP etc.

Git intro

> Git (/ t/)[7] is a distributed version-control system for tracking changes in source code during software development.[8] It is designed for coordinating work among programmers, but it can be used to track changes in any set of files. Its goals include speed,[9] data integrity,[10] and support for distributed, non-linear workflows.[11]

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

- We will introduce Git and Github - commercial Git hosting company `https://en.wikipedia.org/wiki/Git`
- Try creating a Git repository, remember to add a README
- Checkout the repository
- Edit a file
- add and commit it

Use the Github Hello World example: `https://guides.github.com/activities/hello-world/`

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