Software Engineering by ALT-F1

Release v2021.03.23 23.07

Abdelkrim Boujraf

Mar 23, 2021

Contents

Building software requires multiple competencies : understand the business, the regulations, the IT system, the operations, the testing process, the technical debt[1] . . .

By reading this book, you should find sufficient information to manage the manufacturing of software in a systematic method.

Contact us : http://www.alt-f1.be

- OUR COMMITMENT : We strive for superior performance, unmatched work ethic, simple and pragmatic approach, jargon-free language and insightful ideas
- ALT-F1 supports your industry with software, data, analytics & lean optimisations

[1] https://en.wikipedia.org/wiki/Technical_debt

- ALT-F1 designs, implements, deploys and supports secure, large-scale software solutions for diverse industries:
  - Manufacturing
  - MRO Maintenance Repair and Overhaul
  - Warehouse
  - Broadcasting
  - Bank
  - Insurance
  - Defense
  - Automotive
  - Law Enforcement
  - Justice and Serious International Crime

1 Introduction

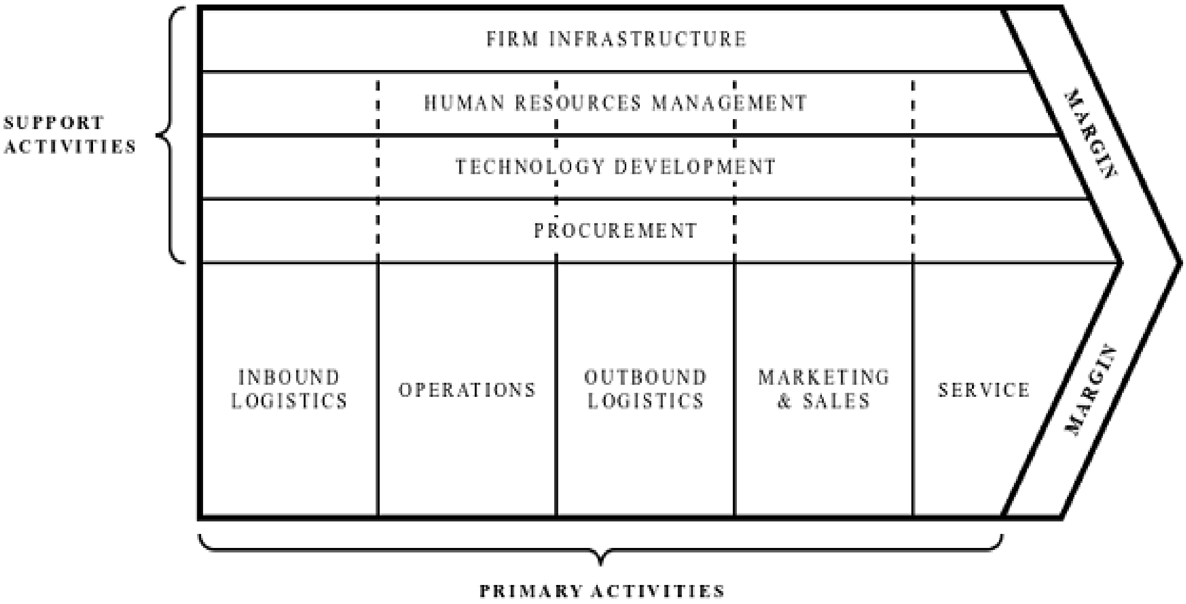
The book is chaptered using the Porter value chain[2] enabling business readers to survive the IT jargon.

Thomas Lee[3] proposed a version of the Porter' value chain adapted to the Software manufacturing: https://www.linkedin.com/pulse/value-chain-software-product-delivery-thomas-lee

```
01-01.Inbound Logistics
01-02.Operations
01-03.Outbound Logistics
01-04.Marketing and Sales
01-05.Services
02-01.Procurement
02-02.Technology development
02-03.HR management
02-04.Firm infrastructure
```

[2] https://s3.amazonaws.com/academia.edu.documents/43857184/Competitive_Advantage-_creative_and_sustaining.pdf?AWSAccessKeyId=AKIAIWOWYYGZ2Y53UL3A&Expires=1556379746&Signature=mJOvChdTYMgqjhGcjKah%2BfMam9Q%3D&response-content-disposition=inline%3B%20filename%3DCompetitive_Advantage-_creative_and_sust.pdf

[3] https://www.linkedin.com/in/tomlee



1.1 Primary activities

Inbound logistics include the receiving, warehousing, and inventory control of input materials.

Operations are the value-creating activities that transform the inputs into the final product.

Outbound logistics are the activities required to get the finished product to the customer, including warehousing, order fulfillment, etc.

Marketing & Sales are those activities associated with getting buyers to purchase the product, including channel selection, advertising, pricing, etc.

Service activities are those that maintain and enhance the product's value including customer support, repair services, etc.

1.2 Support Activities

The primary value chain activities described above are facilitated by support activities. Porter identified four generic categories of support activities, the details of which are industry-specific.

Procurement - the function of purchasing the raw materials and other inputs used in the value-creating activities.

Technology Development - includes research and development, process automation, and other technology development used to support the value-chain activities.

Human Resource Management - the activities associated with recruiting,development, and compensation of employees.

Firm Infrastructure - includes activities such as finance, legal, quality management, etc.

Support activities often are viewed as "overhead", but some firms successfully have used them to develop a competitive advantage, for example, to develop a cost advantage through innovative management of information systems.

Source : http://www.quickmba.com/strategy/value-chain

2 Inbound Logistics

Inbound logistics include the receiving, warehousing, and inventory control of input materials.

Todo: How to retrieve the assets required to build the software : office files, requirements, images, web services definitions

2.1 Speech analogy for Data Vis

Source : https://classroom.udacity.com/courses/ud507/lessons/3063188874/concepts/30639889250923

Parts of speech are to sentences what visual encoding are to charts

POS::Sentences::visual encodings::charts

- Parts of speech are composed of sentences
- charts are composed of visual encodings applied to data types and combined with relationship between those data

Note: data types are continuous or categorical

dimensions are drawn in 1D, 2D, 3D

Geographic charts

- choropleth = geographic + color
- cartogram : geographix + size
- dotmap : georgraphic + shape

2.1.1 The Lie factor

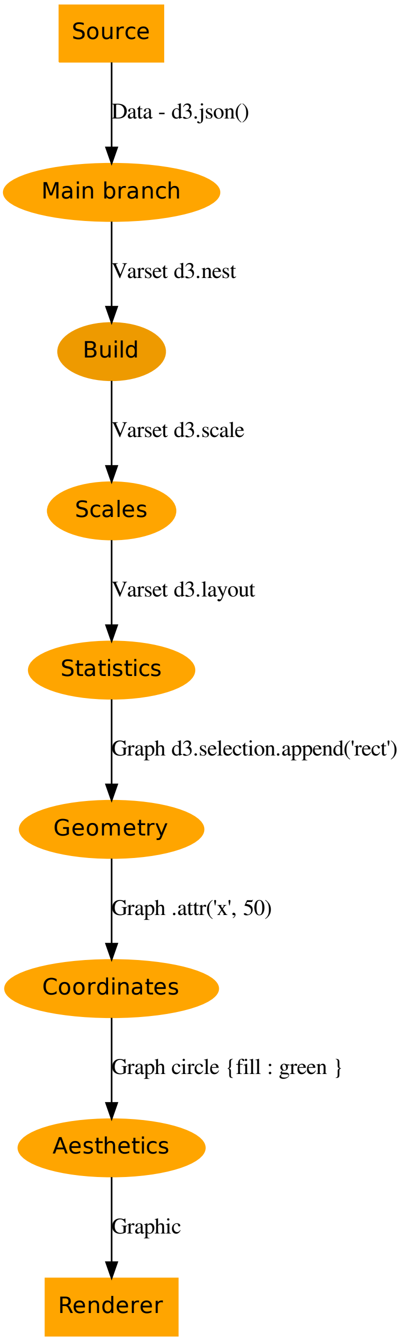
- Lie factor describes the integrity of a graphic. if the lie factor is comprised within [0.95 < lie factor < 1.05] then the graphic representative of the data.

$$\text{Lie factor} = \frac{\text{size fo the effect shown in the graphic}}{\text{size of the effect shown in the data}}[0.95 < liefactor < 1.05] \tag{2.1}$$

2.1.2 Separation of the visual elements and the structure of data

- transform data without changing visual representation
- allow for collaboration across teams

2.1.3 Grammar of the Graphics pipeline



Grammar of Graphics pipeline v2019-05-21

A. d3.layout : applies common transformations on predefined chart objects

B. d3.nest : groups data based on particular keys and returns an array of JSON

C. d3.selection.attr : changes a characteristic of an element such as position or fill

D. d3.json : loads a data file and returns an array of Javascript objects

E. d3.selection.append : inserts HTML or SVG elements into a web page

F. d3.scale : converts data to a pixel or color value that can be displayed

2.1.4 Process

- Pre-attentive processing : https://en.wikipedia.org/wiki/Pre-attentive_processing
- Common chart types and how to choose a chart? https://youtu.be/xD2_AU6atqA

2.1.5 Choosing the right chart

- Chart chooser tool : http://labs.juiceanalytics.com/chartchooser/index.html
- Graph selection matrix : http://www.perceptualedge.com/articles/misc/Graph_Selection_Matrix.pdf
- Visualization types : https://guides.library.duke.edu/datavis/vis_types
- When to use stacked bar charts ? https://solomonmg.github.io/blog/2014/when-to-use-stacked-barcharts/
- Box plots explained : http://www.physics.csbsju.edu/stats/box2.html
- Selecting the Right Graph for Your Message by Stephen Few :
  - http://www.storytellingwithdata.com/blog/2013/04/chart-chooser
  - http://www.perceptualedge.com/articles/ie/the_right_graph.pdf

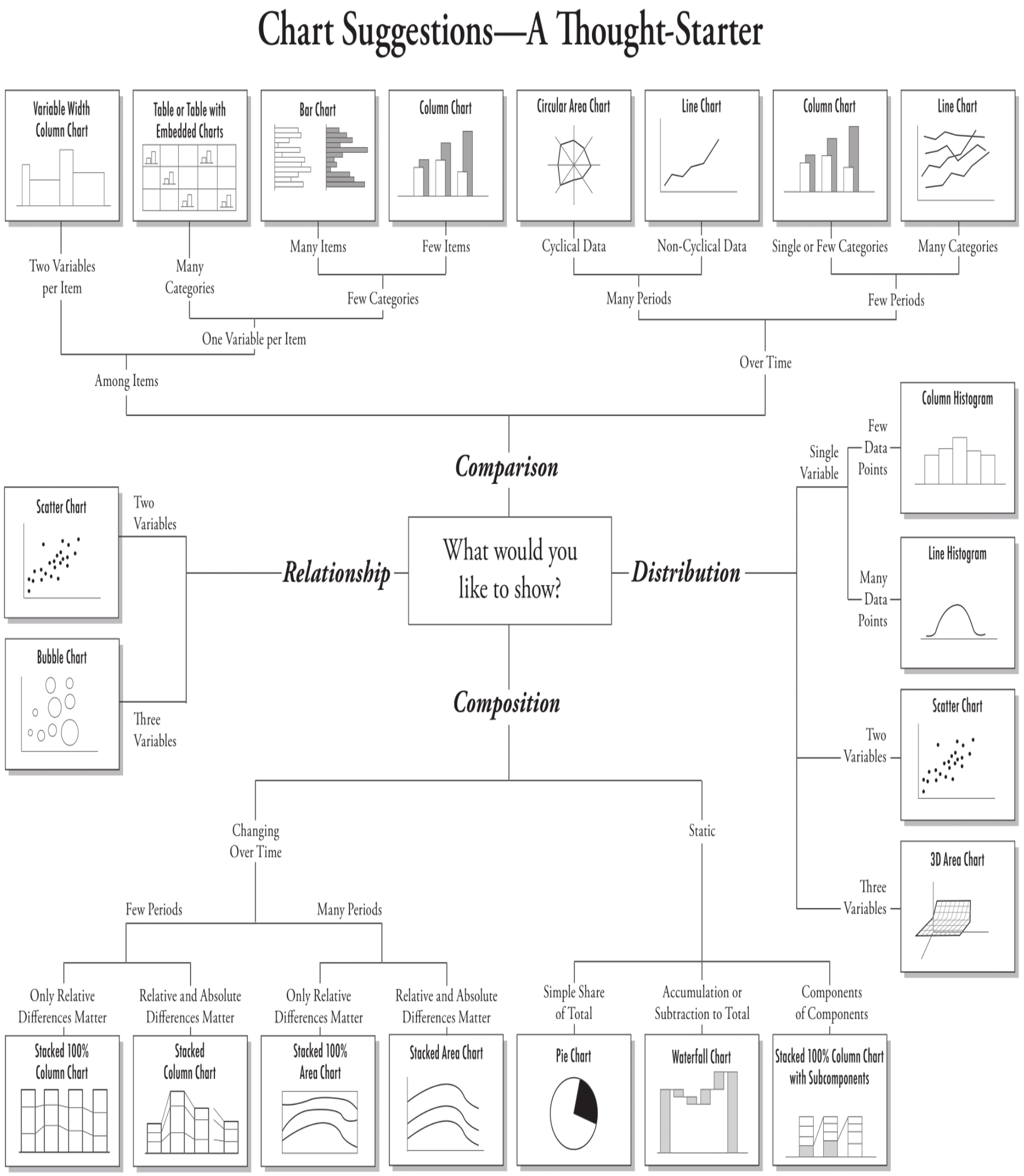


Chart Suggestions—A Thought-Starter

www.ExtremePresentation.com
© 2009 A. Abela — a.v.abela@gmail.com

2.1.6 Choose free tools to draw charts

- https://dimplejs.org
- https://d3js.org
- https://plot.ly
  - US civilian unemployment : https://plot.ly/~Jay-Oh-eN/1
- https://public.tableau.com/s
- https://rawgraphs.io
- https://observablehq.com

- http://openrefine.org
- https://bl.ocks.org

2.1.7 Reviziting the receipt

Source : https://twitter.com/DataToViz/status/1124752405973782528

3 Operations (PM, CI-CD, Backup, Engineering)

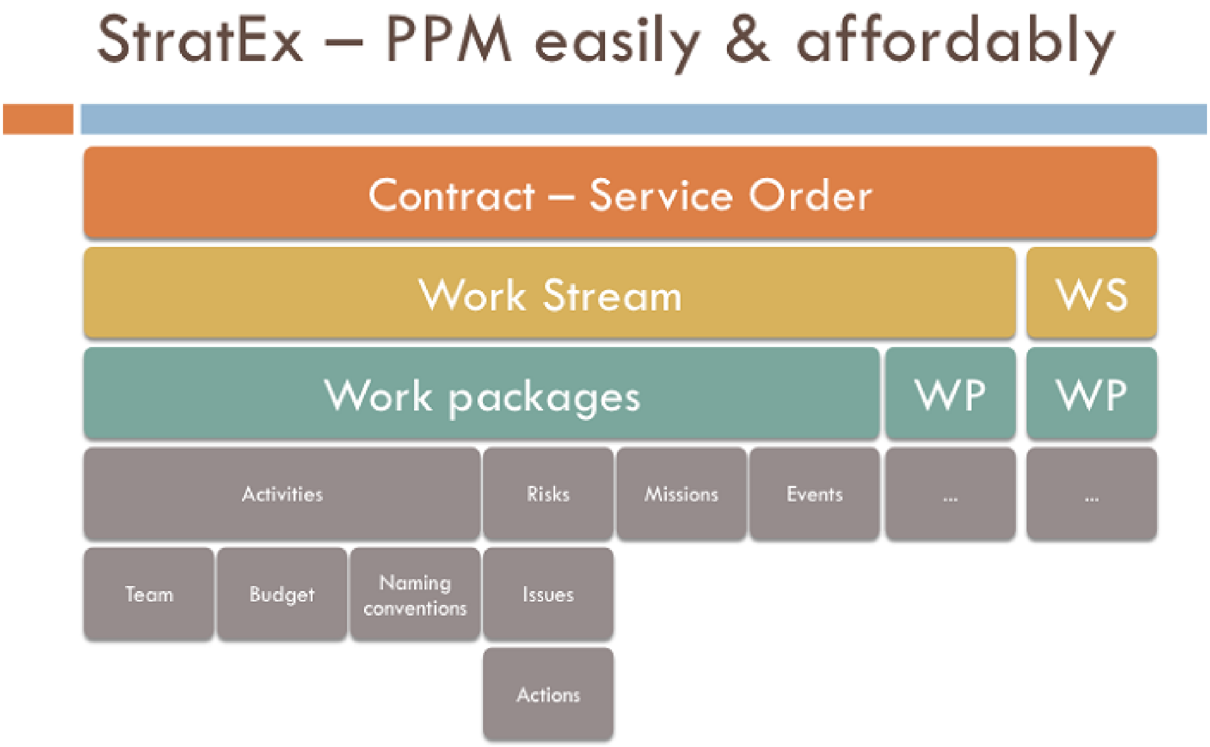
Operations are the value-creating activities that transform the inputs into the final product.

Application deployment and operations required to build a software.

3.1 Project management

StratEx is a web application enabling Managers to control the delivery using customer' processes enforced by conventions shared amongst the team members.

- StratEx App: https://www.stratexapp.com
- StratEx App documentation : https://doc.stratexapp.com



StratEx – PPM easily & affordably

3.2 Programme Management

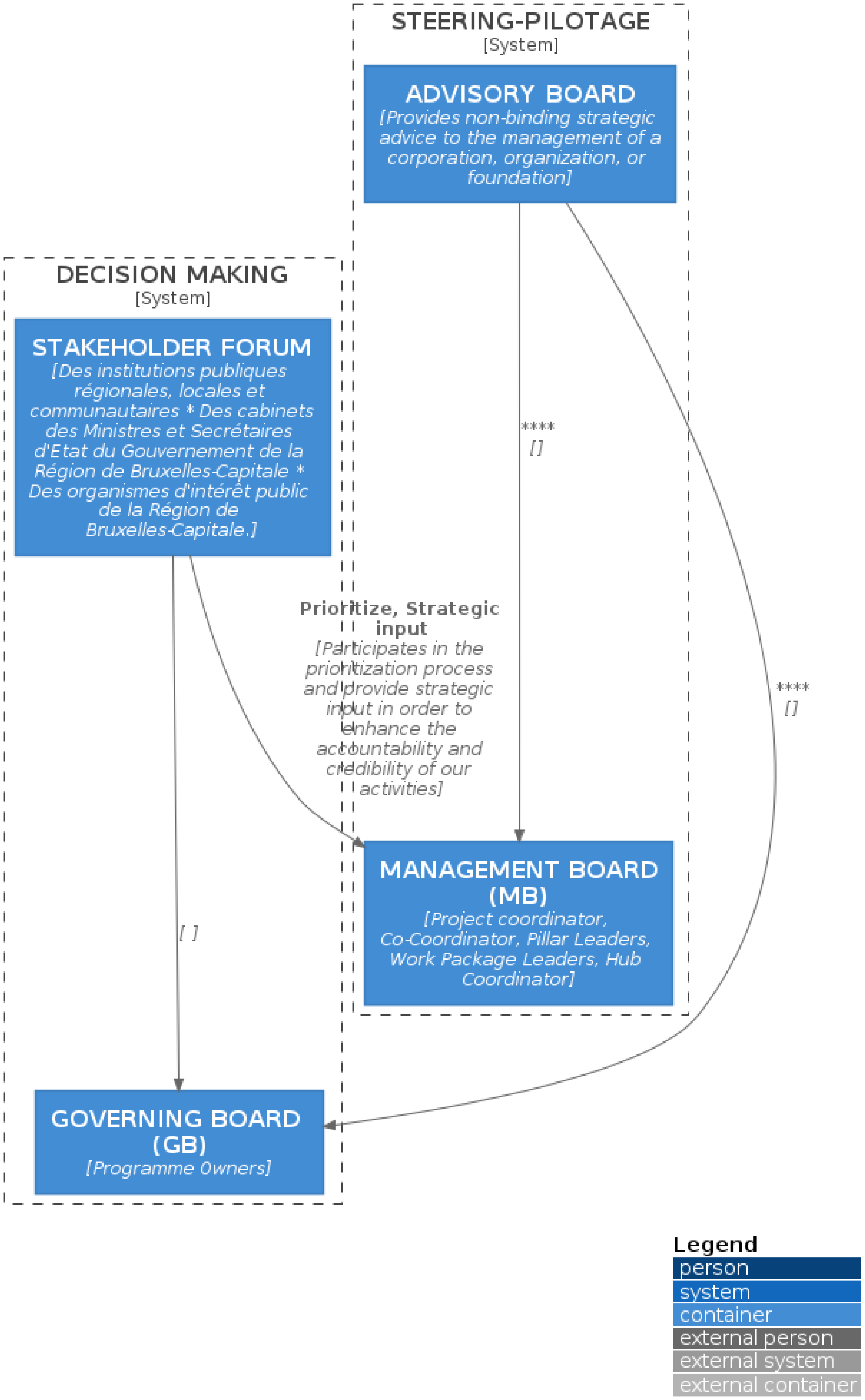
Program Manager Responsibilities

- Organizing programs and activities in accordance with the mission and goals of the organization.
- Developing new programs to support the strategic direction of the organization.
- Creating and managing long-term goals.
- Developing a budget and operating plan for the program.
- Developing an evaluation method to assess program strengths and identify areas for improvement.
- Writing program funding proposals to guarantee uninterrupted delivery of services.
- Managing a team with a diverse array of talents and responsibilities.
- Ensuring goals are met in areas including customer satisfaction, safety, quality, and team member performance.
- Implementing and managing changes and interventions to ensure project goals are achieved.
- Meeting with stakeholders to make communication easy and transparent regarding project issues and decisions on services.
- Producing accurate and timely reporting of program status throughout its life cycle.
- Analyzing program risks.
- Working on strategy with the marketing team.

source: Program Manager Job Description:[4]

[4] https://www.betterteam.com/program-manager-job-description

Programme Management explained


STEERING-PILOTAGE
[System]
ADVISORY BOARD
[Provides non-binding strategic advice to the management of a corporation, organization, or foundation]
DECISION MAKING
[System]
STAKEHOLDER FORUM
[Des institutions publiques régionales, locales et communautaires * Des cabinets des Ministres et Secrétaires d'Etat du Gouvernement de la Région de Bruxelles-Capitale * Des organismes d'intérêt public de la Région de Bruxelles-Capitale.]
****
[]
Prioritize, Strategic input
[Participates in the prioritization process and provide strategic input in order to enhance the accountability and credibility of our activities]
****
[]
[ ]
MANAGEMENT BOARD (MB)
[Project coordinator, Co-Coordinator, Pillar Leaders, Work Package Leaders, Hub Coordinator]
GOVERNING BOARD (GB)
[Programme Owners]
Legend
person
system
container
external person
external system
external container

3.3 Software engineering

Note: The Joel Test - 12 Steps to Better Code[5]

1. Do you use source control?
2. Can you make a build in one step?
3. Do you make daily builds?
4. Do you have a bug database?
5. Do you fix bugs before writing new code?
6. Do you have an up-to-date schedule?
7. Do you have a spec?
8. Do programmers have quiet working conditions?
9. Do you use the best tools money can buy?
10. Do you have testers?
11. Do new candidates write code during their interview?
12. Do you do hallway usability testing?

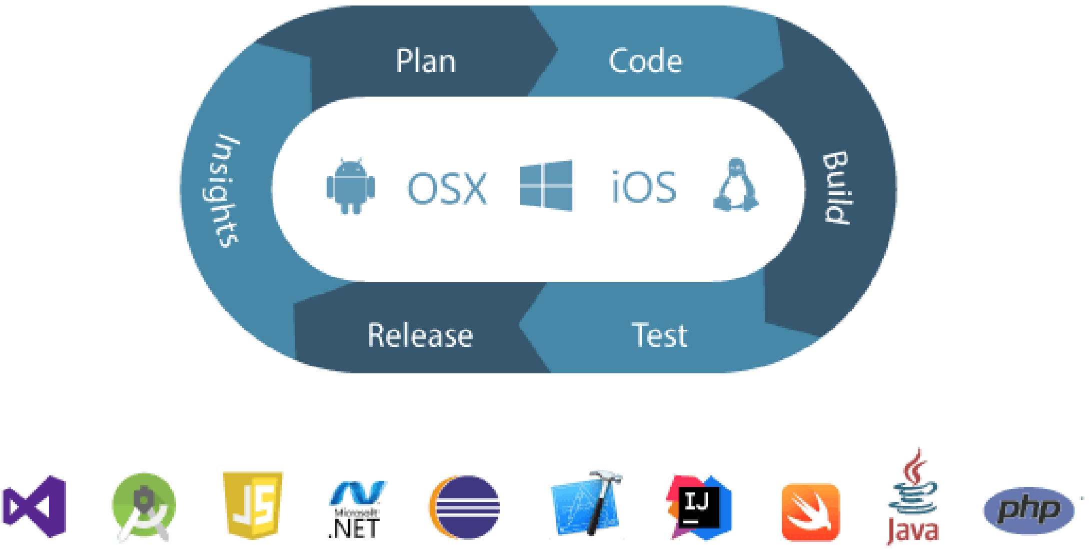
Note: The Simple Programmer Test[6]

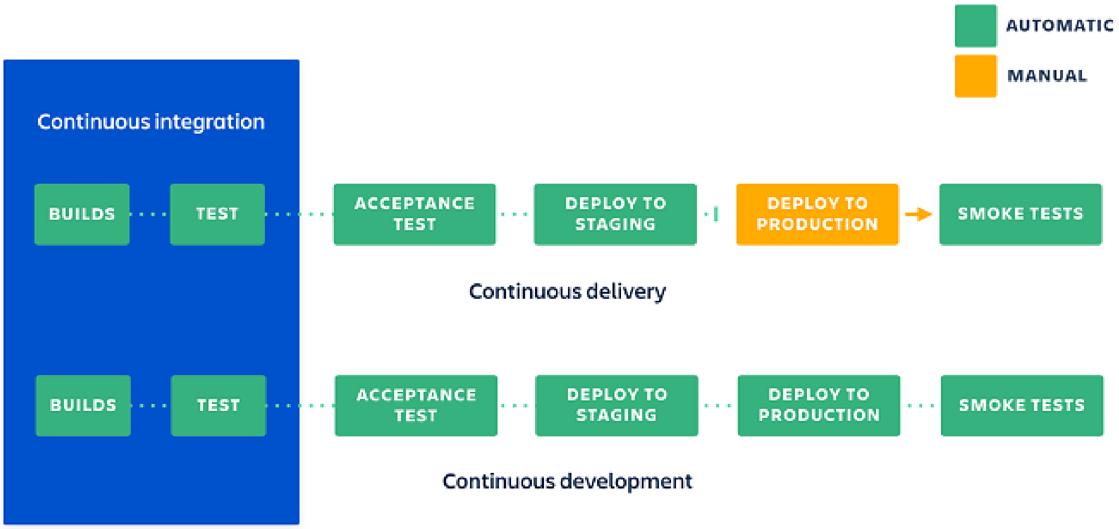
1. Can you use source control effectively?
2. Can you solve algorithm-type problems?
3. Can you program in more than one language or technology?
4. Do you do something to increase your education or skills every day?
5. Do you name things appropriately?
6. Can you communicate your ideas effectively?
7. Do you understand basic design patterns?
8. Do you know how to debug effectively?
9. Do you test your own code?
10. Do you share your knowledge?
11. Do you use the best tools for your job?
12. Can you build an actual application?

[5] https://www.joelonsoftware.com/2000/08/09/the-joel-test-12-steps-to-better-code

[6] https://simpleprogrammer.com/joel-test-programmers-simple-programmer-test

3.4 Continous Integration-Continuous Delivery



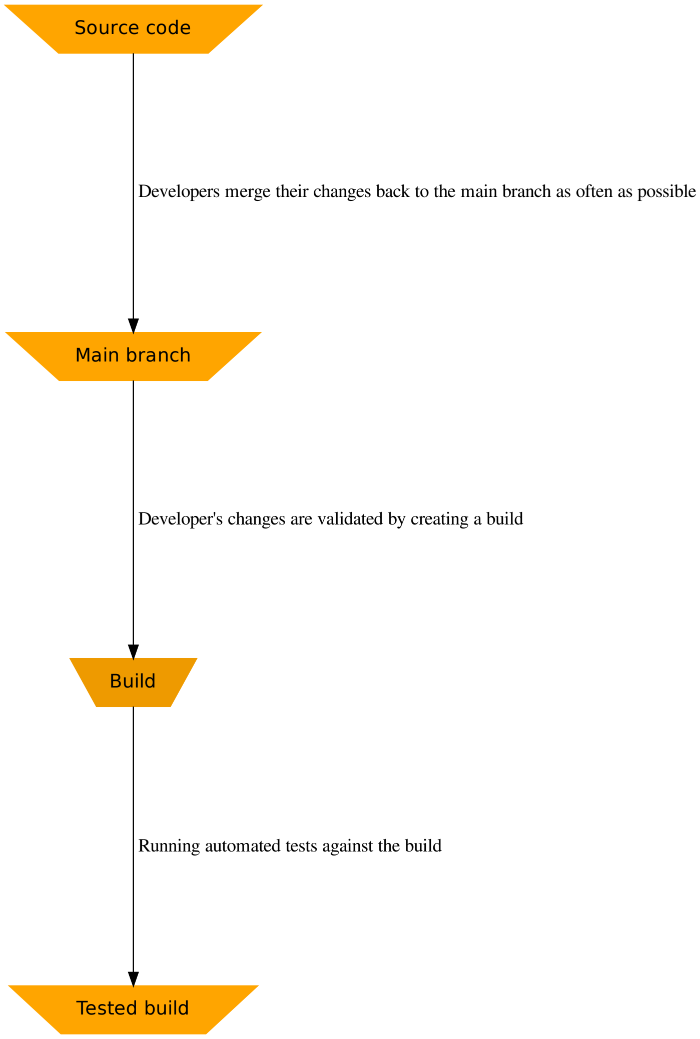


3.4.1 Continuous Integration

Continuous Integration (CI) is straightforward and stands for continuous integration, a practice that focuses on making preparing a release easier. The acronym CD) can either mean continuous delivery or continuous deployment.

source : Continuous integration vs. continuous delivery vs. continuous deployment[7]

[7] https://www.atlassian.com/continuous-delivery/principles/continuous-integration-vs-delivery-vs-deployment


Source code
Developers merge their changes back to the main branch as often as possible
Main branch
Developer's changes are validated by creating a build
Build
Running automated tests against the build
Tested build

Continous Integration v2019-04-27

3.4.2 Continuous Delivery

Continuous delivery (CD) is an extension of continuous integration to make sure that you can release new changes to your customers quickly in a sustainable way.

You automate your release process and deploy your application at any point of time by clicking on a button on a daily, weekly, fortnightly basis.

source : Continuous integration vs. continuous delivery vs. continuous deployment[8]

[8] https://www.atlassian.com/continuous-delivery/principles/continuous-integration-vs-delivery-vs-deployment

3.4.3 Continuous Deployment

Continuous deployment (CD) implies that every change that passes all stages of your production pipeline is released to your customers. There's no human intervention, and only a failed test will prevent a new change to be deployed to production.

source : Continuous integration vs. continuous delivery vs. continuous deployment[9]

3.5 Backup

3.5.1 Which ressources do you need to backup?

- Virtual machines running an Operating system (Windows, MacOSX, Linux. . .)
- Databases (Microsoft SQL Server, MongoDB, Oracle Database, MySQL. . .)
- File systems (Directories containing documents, images. . .)

3.5.2 Which strategy do you need to apply?

- Store forever one full backup per year from January the 1st (Year Y1, Y2. . .)
- Store everyday day one incremental backup for each resource
  - Store a full backup the first of each month
  - Store an incremental backup from Day 2 to the end of the month (31-1 backups)
  - Replace the incremental backup performed one month earlier (Month M-1)
  - Keep the full backup made once a month

3.5.3 Wrap up

- Store in total :
  - 12 full backups per year for each month (1st day of each month)
  - 31 incremental backups for the last 31 days
  - 1 full backup per year
- Example:
  - Year 1 : 12 + 31 backups
  - Year 2 : 12 + 31 + 1 backups
  - Year 3 : 12 + 31 + 2 backups

4 Outbound Logistics (Deployment CI/CD)

Outbound logistics are the activities required to get the finished product to the customer, including warehousing, order fulfillment, etc.

Todo: How to deliver the software : package, virtual machine, deployed application on the cloud

[9] https://www.atlassian.com/continuous-delivery/principles/continuous-integration-vs-delivery-vs-deployment

5 Marketing and Sales

Marketing & Sales are those activities associated with getting buyers to purchase the product, including channel selection, advertising, pricing, etc.

Todo: tools and tasks required to design the offer, market the product and then sell it offline or online

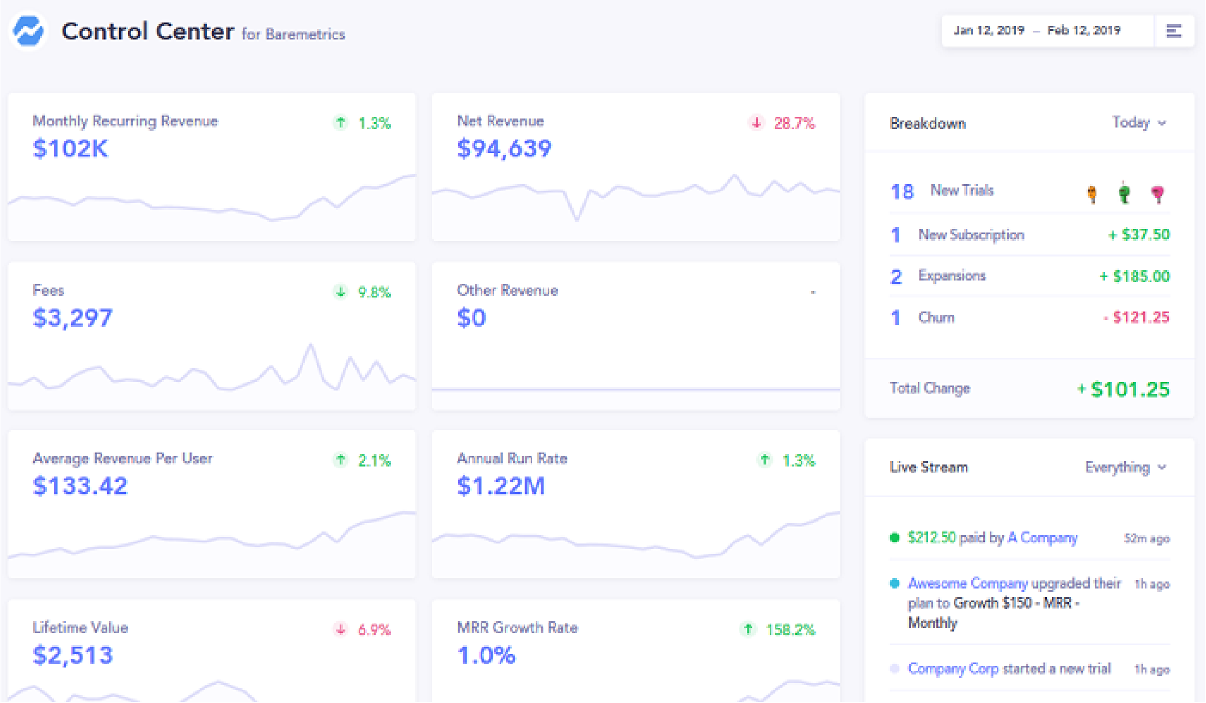
5.1 Sales automation

- Use Zoho CRM : https://www.zoho.com/crm/

1. Créer des **leads** (nom, prénom, email address, nom de société) pour XYZ digital au minimum
2. Convertissez des **leads** en **Contact** (personne physique travaillant pour une personne morale) et **Account** (personne morale)
3. Ajouter un **deal** lorsque vous convertissez un **lead** en **Contact**
4. Créer des **deals** (projets signés ou en cours)

5.2 Sales Dashboard

Baremetrics : https://demo.baremetrics.com



Control Center for Baremetrics

Jan 12, 2019 – Feb 12, 2019

Monthly Recurring Revenue ↑ 1.3%
$102K

Net Revenue ↓ 28.7%
$94,639

Fees ↓ 9.8%
$3,297

Other Revenue -
$0

Average Revenue Per User ↑ 2.1%
$133.42

Annual Run Rate ↑ 1.3%
$1.22M

Lifetime Value ↓ 6.9%
$2,513

MRR Growth Rate ↑ 158.2%
1.0%

Breakdown Today

18	New Trials	
1	New Subscription	+ $37.50
2	Expansions	+ $185.00
1	Churn	- $121.25

Total Change + $101.25

Live Stream Everything

● $212.50 paid by A Company 52m ago

● Awesome Company upgraded their plan to Growth $150 - MRR - Monthly 1h ago

● Company Corp started a new trial 1h ago

5.3 Contracts management

- The Service Provider
  - **is** Shareholder of the customer
    - the Service Provider Lowers the price
  - **is not** Shareholder of the customer
    - the Service Provider asks for a regular price
- The contract includes
  - 3rd parties with whom the Service Provider **has no** liabilities towards the 3rd party
    - The Service Provider has NOTHING to do contractually
  - 3rd parties with whom the Service Provider **has** liabilities towards the 3rd party
    - A contract describing the governance MUST be written

6 Services

Service activities are those that maintain and enhance the product's value including customer support, repair services, etc.

Todo: Tools required to support the customer during the usage of the software such as https://readthedocs.org

7 Procurement

Procurement - the function of purchasing the raw materials and other inputs used in the value-creating activities.

Todo: How to buy efficiently the software, computers, network devices, IoT devices you use to build your solutions

8 Technology development (Tools Git, ngrok)

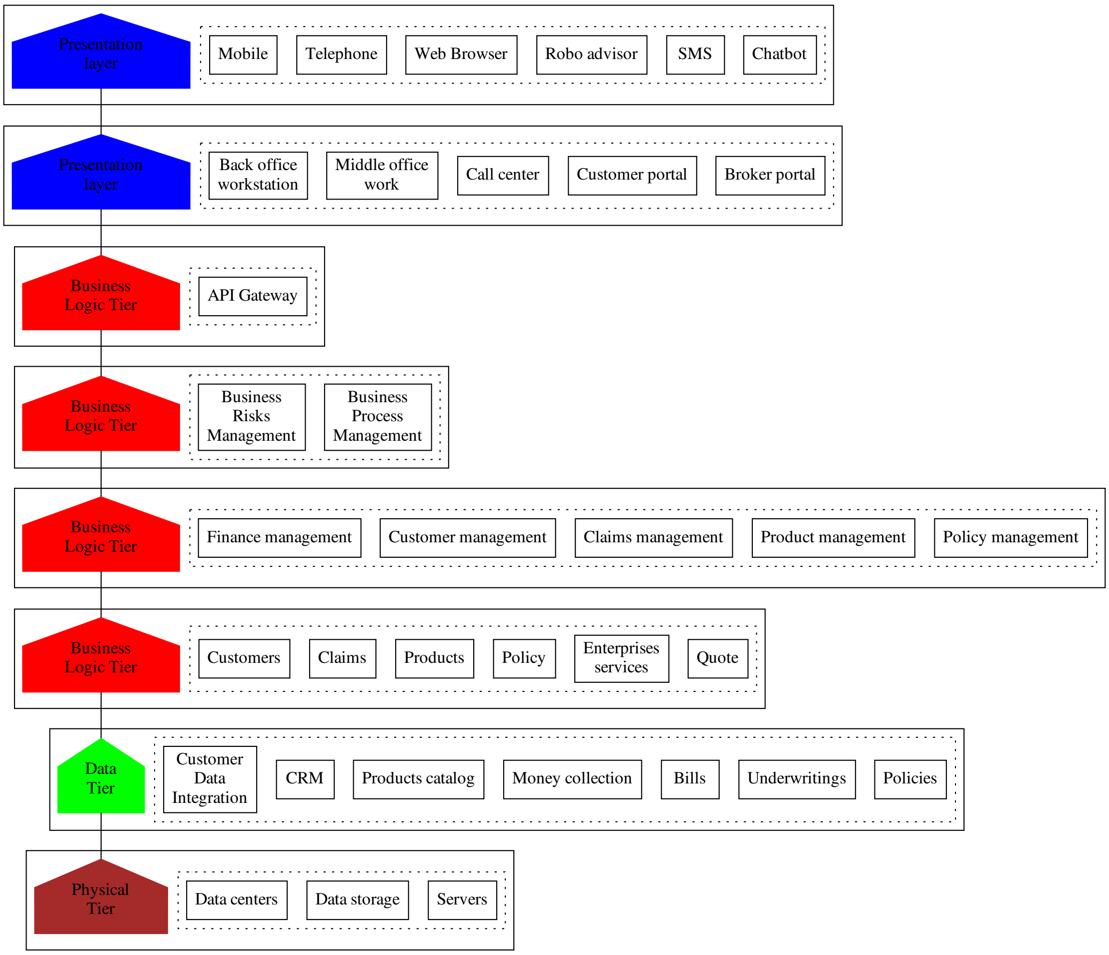
Technology Development - includes research and development, process automation, and other technology development used to support the value-chain activities.

8.1 The digitization

The *Digitization* is *"Digitization, less commonly digitalization, is the process of converting information into a digital (i.e. computer-readable) format, in which the information is organized into bits."*

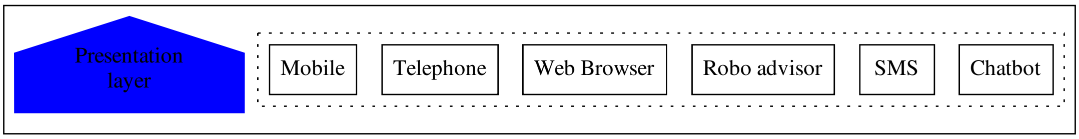
In short, developing software mimicking manual processes while adding new capabilities such enabling the communication between IT systems from different companies that couldn't have been integrated using another way.

8.1.1 Digitization for the Insurance industry



Digitization of the Insurance industry v2019-09-12

Presentation layer I



The presentation layer is the layer that is in connection with your customers through several means or devices :

Telephone

A service desk supporting the requests on the phone

Robo advisor

A *Robo advisor is a class of financial adviser that provide financial advice or Investment management online with moderate to minimal human intervention*

- The *Robo advisor* scans and dematerializes your documents
- The *Robo advisor* compares the available offers and optimize the portfolio of your customers
- The customer interacts with the trusted Robo advisors via the web, mobile chat or email

Mobile or Smartphone

Mobile phones accessing the insurer' services through SMS, email, voice of mobile app means

SMS

The customer interacts with the Insurer using SMS. The answers to the SMS are made by a *Robo advisor*, a *Chatbot* or a human. The help desks uses an IT system and (s)he is not necessarily answering using a mobile phone.

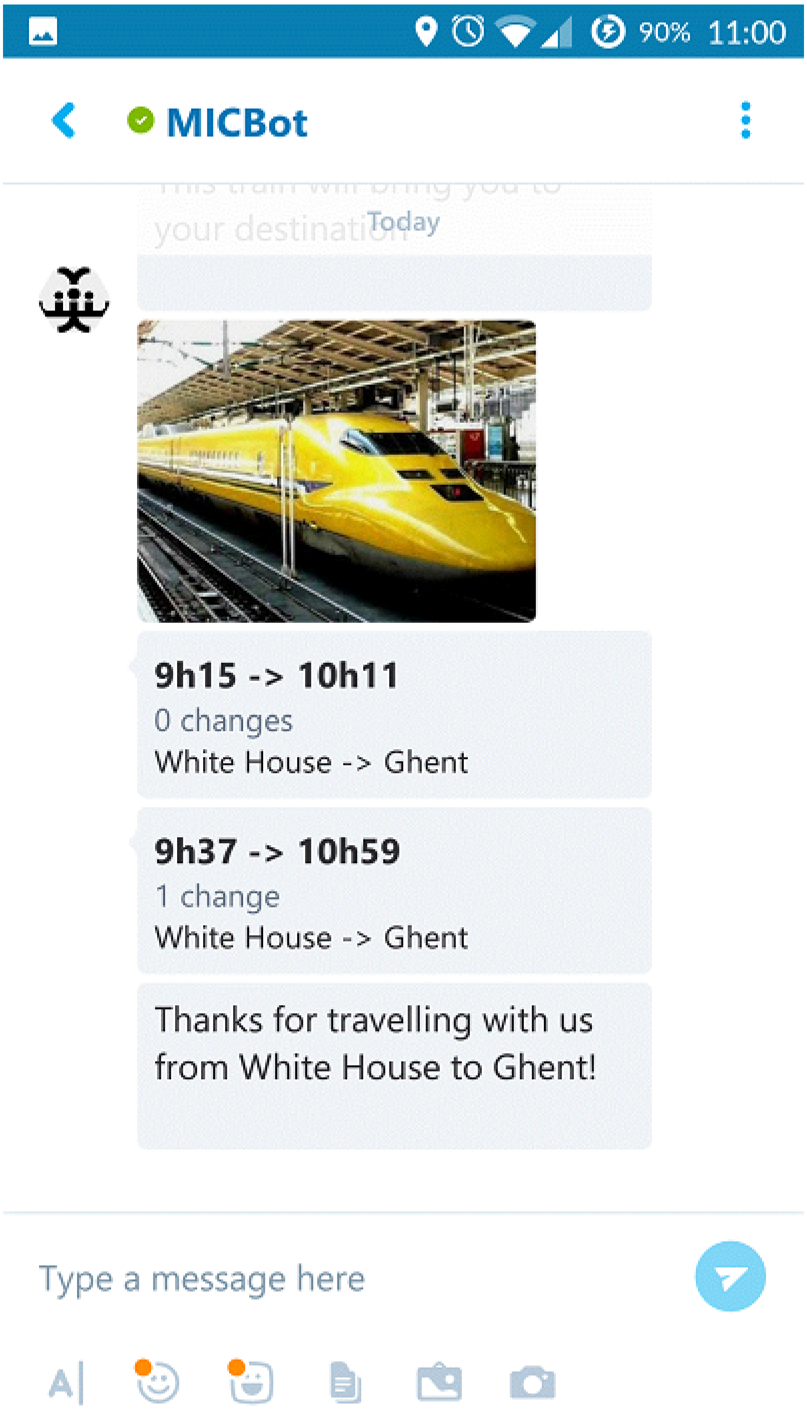
Web Browser

a software used to access the Web site of the insurer or the *Underwriter* (i.e. Firefox, Opera, Ecosia, Microsoft Internet Explorer, Google Chrome, Microsoft Edge, Safari)

Chatbot

The *Chatbot* is a piece of software that conducts a conversation via auditory or textual methods.

The *Chatbot* tries to answer customers' questions that human would have had as a conversation. Some chatbots use sophisticated natural language processing systems but are most of time supported by a service desk run by humans when the *Chatbot* can't understand the demands made by the customer.



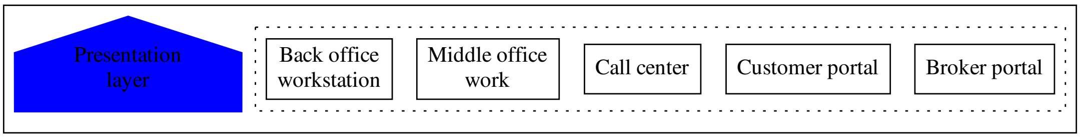
This train will bring you to your destination

9h15 -> 10h11
0 changes
White House -> Ghent

9h37 -> 10h59
1 change
White House -> Ghent

Thanks for travelling with us from White House to Ghent!

Presentation layer II



The presentation layer counts the front office as well as the back office. The *Back office is all the resources of* *the company that are devoted to actually producing a product or service and all the other labor that isn't seen by customers, such as administration or logistics.*

Back office workstation

Those are the collaborators managing the operations ensuring the correct exection of the processes :

- quote an offer
- validate personal data
- validate the filling of a form
- perform the dunning services duties

The back office requires documentation, software and reports to perform their duties.

Middle office workstation

The *Middle office is made up of the risk managers and the information technology managers who manage risk and maintain the information resources.*

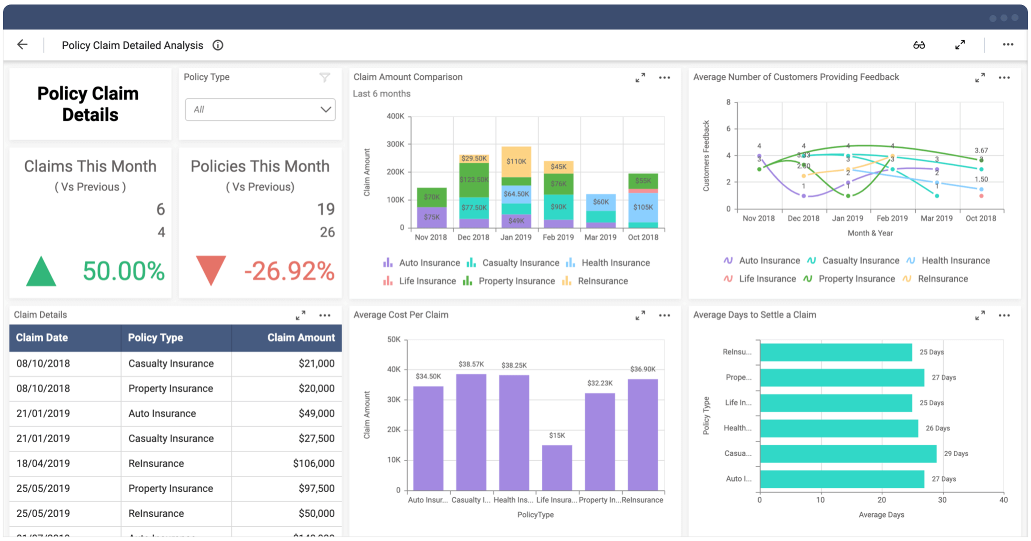
- Track claim settlement times
- Customer satisfaction ratings
- Long-term trends in customer activity

Data collected during the operations are stored into IT systems operated by a multitude of managers (risks, operations, HR, Marketing)

Those data are analysed and supports the value chain (logistics, operations, marketing, sales, support) by giving a broad and exact view of the financial situation of the company.

After a careful analysis, the data are shared with the back office who can act and interact with prospects, customers and suppliers depending on the situation (dunning service, quotation, billing, closing off the contract)

Tip: Try the Insurance claims analysis[10] dashboard



Claims This Month

(Vs Previous)

6

4

▲ 50.00%

Policies This Month

(Vs Previous)

19

26

▼ -26.92%

Claim Details

Claim Date	Policy Type	Claim Amount
08/10/2018	Casualty Insurance	$21,000
08/10/2018	Property Insurance	$20,000
21/01/2019	Auto Insurance	$49,000
21/01/2019	Casualty Insurance	$27,500
18/04/2019	ReInsurance	$106,000
25/05/2019	Property Insurance	$97,500
25/05/2019	ReInsurance	$50,000

[10] https://demo.boldbi.com/bi/en-us/dashboards/ad50e869-4fe8-4e8f-9137-54c3aec5ee54/insurance/insurance%20analysis%20dashboard

Call centre

Three types of call centres might be operated by a financial service company:

- An *Inbound call center* is operated by a company to administer incoming product or service support or information enquiries from consumers.
- An *Outbound call center* is operated for telemarketing, for solicitation of charitable or political donations, debt collection, market research, emergency notifications, and urgent/critical needs blood banks.
- A *Contact center*, further extension to call centers administers centralized handling of individual communications, including letters, faxes, live support software, social media, instant message, and e-mail.

Customer portal

A website accessible through a Web browser or a mobile phone enabling the customer to access all the aspects of his duties and rights towards the insurer.

The portal gives access to diverse functionalities:

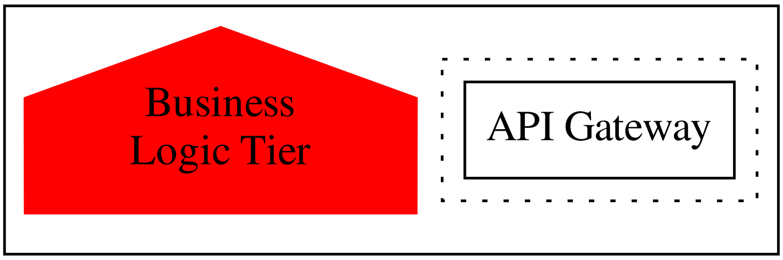
- Information platform: share details about the products and services, how to contact the insurer
- Transaction platform: create, update or delete information, stop a current insurance, pay electronically the remaining Bills
- Sales platform: generate up-sell and cross-sell opportunities, promote the *Robo advisor* capabilities
- Rewards platform: Insurers retain their customers through the Perceived Value of the customer, the Affinity that the customer has with his insurer, and the Barriers to Exit
  - Perceived value: does the customer feels that (s)he has coverage at a competitive and fair price?
  - Affinity: do the customer has a emotional connection with the customer? Insurance products may tend to have a limited value due to the commoditized nature of the product
  - Barriers to Exit: does the customer has strong and effective reasons to do not leave an insurer? the lack of competition, the increase of costs, the loss of a unique protection, a decrease of the quality of service

Broker portal

A broker portal is a website enabling the *Broker* to perform her/his duties

- Information platform: share details about the products and services, how to contact the insurer, the customers
- Sales platform: support the sales process (from a quote to a signed contract), generate up-sell and cross-sell opportunities
- Marketing platform: identify new sales opportunites by advertising the products and identify the most profitable or potential prospects
- CRM platform: maintain data related to the prospects and customers (contact details, online and offline interactions)
- Dunning service platform: inform and give the tools to enable the broker to run after unpaid invoices till the termination of the contract

Business logic tier I



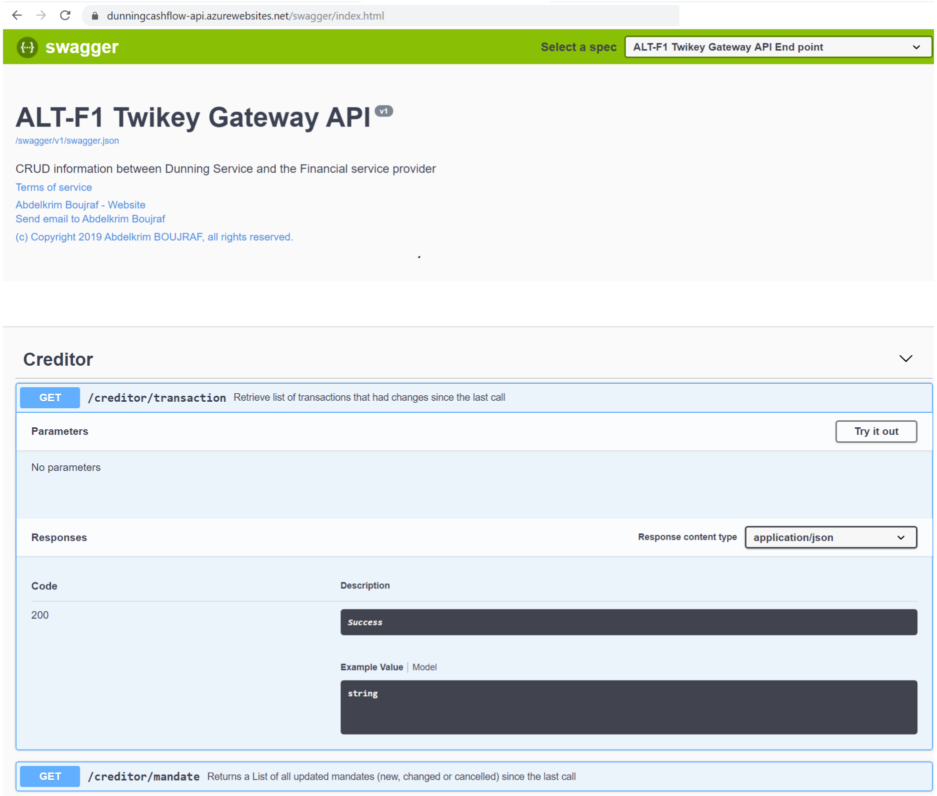
API Gateway

The API (Application Programming Interface) describes the functions or the interfaces available between a client and a server.

APIs are enablers of the platform economy, and allow users to enhance and add services over existing products.

For example: An API enables an application ‘A’ to query a system ‘B’ and collects the schedule of the public transportation (See https://opendata.stib-mivb.be/store/data)

Tip: Look at the description of the API from a dunning Service https://dunningcashflow-api.azurewebsites.net/swagger/index.html


dunningcashflow-api.azurewebsites.net/swagger/index.html
swagger
Select a spec
ALT-F1 Twikey Gateway API End point
v1
Try it out
Example Value
Model

ALT-F1 Twikey Gateway API

/swagger/v1/swagger.json

CRUD information between Dunning Service and the Financial service provider

Terms of service

Abdelkrim Boujraf - Website

Send email to Abdelkrim Boujraf

(c) Copyright 2019 Abdelkrim BOUJRAF, all rights reserved.

Creditor

GET `/creditor/transaction` Retrieve list of transactions that had changes since the last call

Parameters

No parameters

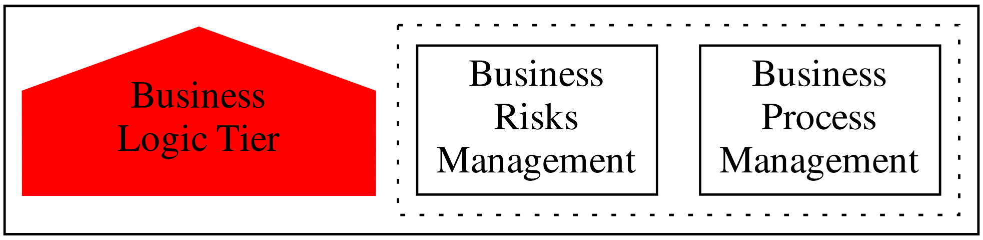
Responses

Response content type: application/json

Code	Description
200	*Success* Example Value | Model `string`

GET `/creditor/mandate` Returns a List of all updated mandates (new, changed or cancelled) since the last call

Business logic tier II



Business Risk Management

The financial services must comply with a multitude of risks.

Here are a list of pure risks (loss or no loss only) that an insurer or a *Underwriter* may be confronted with:

- Regulatory Compliance: Invoice compliance, MiFID ii, MiFID 2[11], Solvency II, Solvency 2[12]
- Tax Compliance: Tax determination, Fiscal reporting, VAT reporting
- Liability risk exposure: product liability risks, or contractual liability risks
- Operational risk: mistakes in process and procedure
- Intellectual property violation risk
- Mortality and morbidity risk at the societal and global level

Warning:

- Speculative risks are not described in this documents. i.e. market risk, reputational risk, brand risk, product success risk. . .

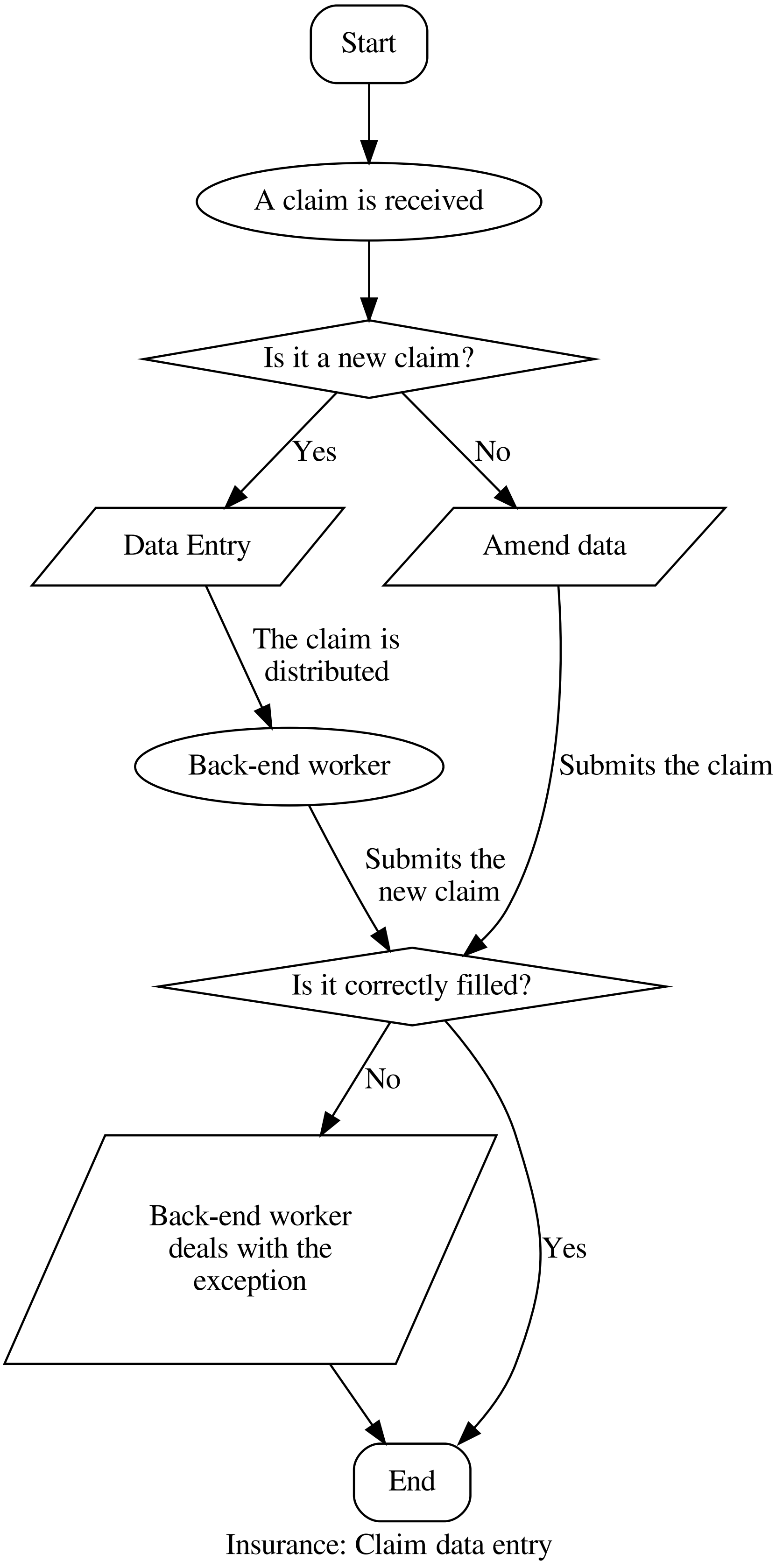
Business Process Management

The *Business Process Management is a discipline aimed at managing all aspect of the business processes; from process design to modeling and analysis to execution and improvement.*

Note: Here is the description of a process: Data entry of a claim

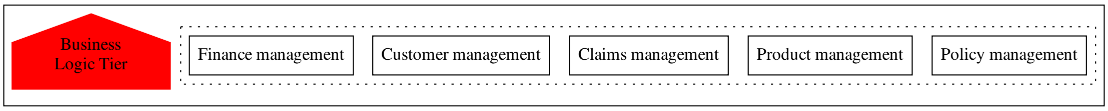
[11] https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=celex%3A32014L0065

[12] https://ec.europa.eu/info/law/markets-financial-instruments-mifid-ii-directive-2014-65-eu_en



Insurance: Claim data entry

Business logic tier III



Finance management

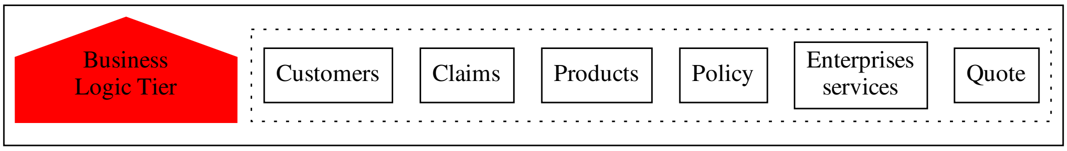
Customer management

Claims management

Product management

Policy management

Business logic tier IV



Customers

Claims

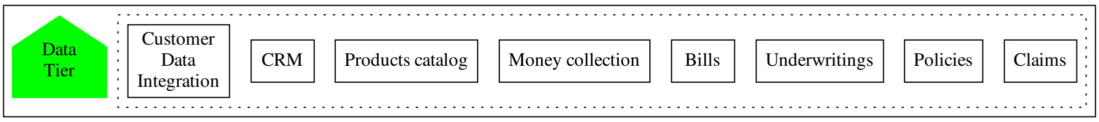
Products

Policies

Entreprise services

Quotes

Data tier I



CDI (Customer Data Integration)

CDI (Customer Data Integration)

CRM (Customer Relationship Management)

CRM (Customer Relationship Management)

Products catalog

Money collection

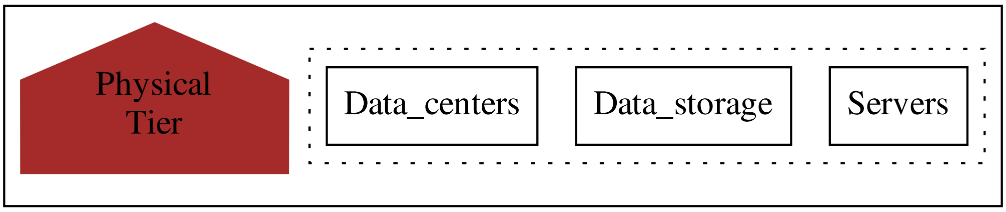
Bills

Underwritings

Policies

Claims

Physical tier



Data Centers

Data storage

Servers

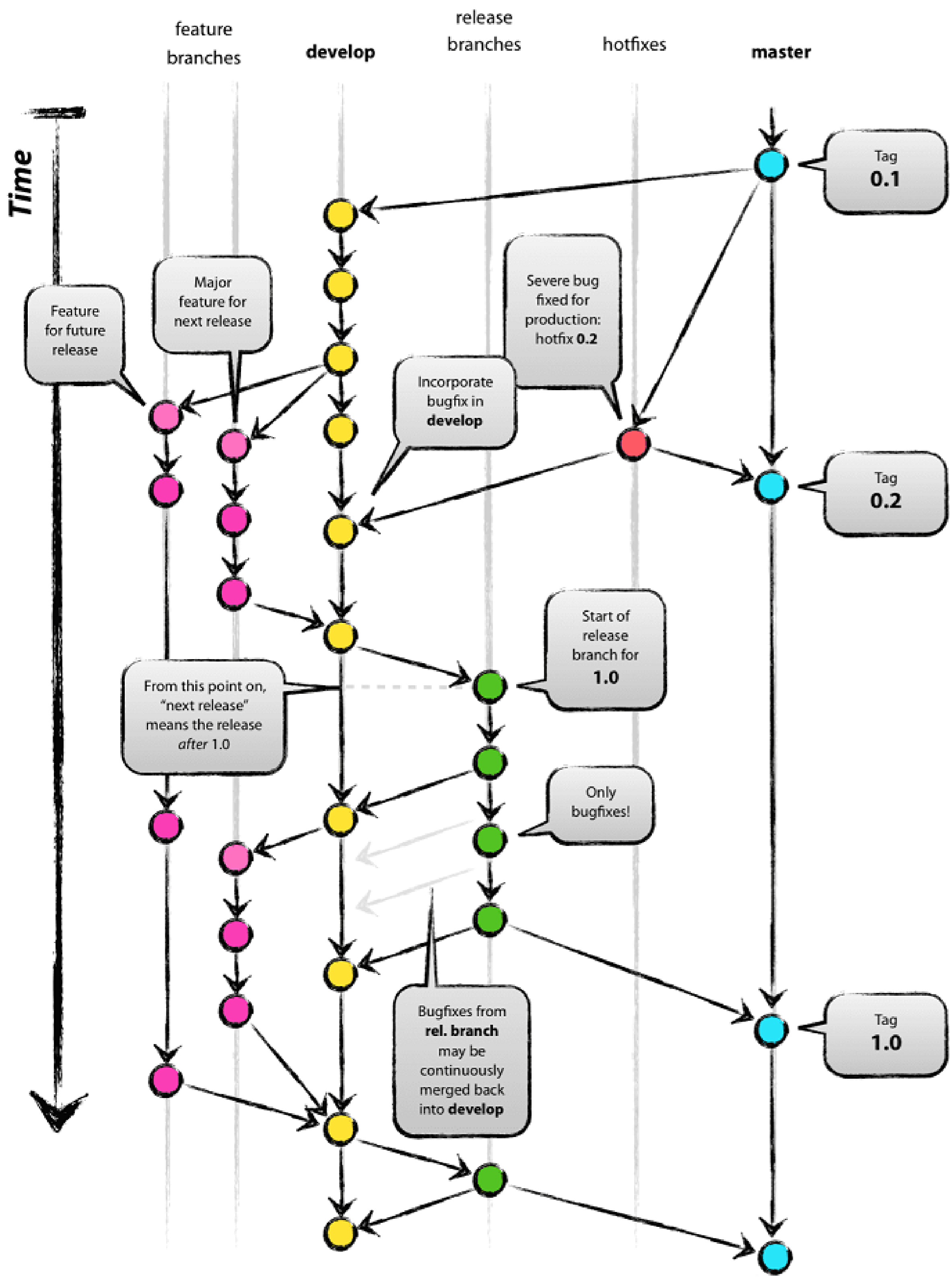
8.2 GIT lifecycle

Description of how to manage the versions, branches in a git repository as well the operations of the software

How to write relevant commits?

8.2.1 A successful git branching model

A successful Git branching model : https://nvie.com/posts/a-successful-git-branching-model/


feature branches
develop
release branches
hotfixes
master
Time
Tag 0.1
Severe bug fixed for production: hotfix **0.2**
Major feature for next release
Feature for future release
Incorporate bugfix in **develop**
Tag 0.2
Start of release branch for 1.0
From this point on, "next release" means the release *after* 1.0
Only bugfixes!
Bugfixes from **rel. branch** may be continuously merged back into **develop**
Tag 1.0

8.2.2 GIT : commit conventions

source : conventional commits : https://github.com/conventional-commits

Semantic messages: http://seesparkbox.com/foundry/semantic_commit_messages

```
build
chore (maintain i.e. updating grunt tasks etc; no production code change)
ci (continuous integration)
docs (documentation)
feat (feature)
fix (bug fix)
perf (performance improvements)
refactor (refactoring production code)
revert
style (formatting, missing semi colons, ...)
test (adding missing tests, refactoring tests; no production code change)
```

- enumeration : https://github.com/conventional-changelog/commitlint/tree/master/%40commitlint/config-conventional#type-enum
- Install commitlint : https://conventional-changelog.github.io/commitlint/#/

Example of commits

Source : https://github.com/conventional-commits/conventionalcommits.org/commits/master

8.2.3 GIT : how to manage the versions, branches . . . ?

- **GIT** : Create a branch : [BRANCH-DEV] – [BRANCH-PARENT]
- **DEV** : Local development on Software engineer machine
- **MERGE GIT** : Merge the [BRANCH-DEV] with the [BRANCH-PARENT]
  - The code is merged into the [BRANCH-DEV]
- **STAGING** : The Software is deployed on the staging environment
- **MERGE GIT** : Merge the [BRANCH-DEV] with the [BRANCH-PARENT]
  - The code is merged into the [BRANCH-PARENT]
- **PROD** : Test the PROD version of the software
- **LIVE** : deploy the PROD version of the software on the PROD server

8.2.4 GIT LFS Large File System

- git commands
- Install git lfs https://git-lfs.github.com
- **Locks**
  - git lfs lock images/foo.jpg
  - git lfs locks
  - git lfs unlock images/foo.jpg
- git lfs push origin master –all

Create a .gitattributes file

```
.. include:: .gitattributes
```

Commands to add files into the repository, and push the code

```
git lfs install
git lfs track "*.jpg" --lockable
git lfs track "*.JPG" --lockable
git lfs track "*.png" --lockable
git lfs track "*.zip" --lockable
git lfs track "*.mp4" --lockable
git lfs track "*.MP4" --lockable
git lfs track "*.docx" --lockable
git lfs track "*.svg" --lockable
git lfs track "*.gif" --lockable
git lfs track "*.psd" --lockable
git lfs track "*.sketch" --lockable
git lfs track "*.ai" --lockable

git add "*.jpg" "*.JPG" "*.png" "*.zip" "*.mp4" "*.MP4" "*.docx" "*.svg" "*.gif"

git lfs ls-files
git lfs env

git config lfs.https://inlsprl.visualstudio.com/[ProjectName]/_git/[ProjectName].
↪git/info/lfs.locksverify true
git push origin master
git lfs push origin master --all
```

- https://github.com/git-lfs/git-lfs/wiki/File-Locking

8.3 ngrok - Public URLs for exposing your local web server

Use ngrok to grant access to your localhost to anyone

1. Install https://ngrok.com/download
2. open the port where the web server is located. run the following command :
    1. **[path to ngrok]**\ngrok.exe http **[port to open on your localhost]** -host-header=rewrite
3. share the URL to the person who needs to access your local machine. i.e. https://a1cc816e.ngrok.io

```
ngrok by @inconshreveable

Session Status online Account [the account name] (Plan: Free) Update update␣
↪available (version 2.2.8, Ctrl-U to update) Version 2.2.3
Region United States (us) Web Interface http://127.0.0.1:4040
Forwarding http://a1cc816e.ngrok.io -> localhost:4624 Forwarding

https://a1cc816e.ngrok.io -> localhost:4624
```

8.4 How to write a bug report?

1. Copy paste the content hereunder
2. Create a new issue: https://bitbucket.org/altf1be/software-architecture/issues/new

```
## WHAT STEPS WILL REPRODUCE THE PROBLEM?

1. Open the page
2.
3.

## WHAT IS THE EXPECTED OUTPUT?

*  StratEx is loaded


## WHAT DO YOU SEE INSTEAD?

* The screenshot attached to this email
* StratEx cannot be opened because of a problem

## WHAT VERSION OF THE PRODUCT ARE YOU USING?

* Version: 3.5.6245.20028
* on [https://www.stratexapp.com](https://www.stratexapp.com)
* on [https://staging.stratexapp.com](https://staging.stratexapp.com)
* on [https://develop.stratexapp.com](https://develop.stratexapp.com)


## ON WHAT OPERATING SYSTEM, BROWSER, ETC.?

* Windows 7.1
   * Chrome Version 54
   * Internet Explorer 11
   * Opera Version 41
* Windows 10
   * Internet Explorer
   * Edge
* Mac OS X 10.9 (13A603)
   * Safari Version 7.0 (9537.71)
   * Chrome Version 31.0.1650.57

## PLEASE PROVIDE ANY ADDITIONAL INFORMATION BELOW.

* None
* Extra files are available on [StratExApp files on Google Drive]
* Find the private [Videos generated on GDrive]
* Find the public [Videos on StratEx YouTube channel]
* Find the public [Documentation on Read The Docs]

## Bug report (if any)

* None

[StratExApp files on Google Drive]: https://drive.google.com/a/alt-f1.be/
↪folderview?id=0B9L2cx0TUjLGUFZBSkF6WlFCYms&usp=sharing#list
[Videos generated on GDrive]: https://drive.google.com/a/alt-f1.be/folderview?
↪id=0B9L2cx0TUjLGa190N1ZURHBpUFE&usp=sharing
[Videos on StratEx YouTube channel]: https://www.youtube.com/channel/
↪UCuwGfoVoozq0ZTmHJ3WCvTQ
[Documentation on Read The Docs]: http://stratexapp-docs.readthedocs.org/en/latest/
```

8.5 Research & Development topics

0. Prerequisite:
   - Use Technologies supported by Microsoft
     - ASP.NET - https://docs.microsoft.com/en-us/aspnet/overview
     - ASP.NET Core - https://docs.microsoft.com/en-us/aspnet/core/
     - Microsoft Azure: https://portal.azure.com
1. Single Page Application (SPA)
   - Build a SPA such as Microsoft Azure for our customers
   - Test SPA App
2. IT Automation best practices
   - Reading : https://www.joelonsoftware.com/2000/08/09/the-joel-test-12-steps-to-better-code/
   - In general we should go for 1-click test/deployment
   - Design
   - Development
   - Test (SPA, Web apps)
     - Unit
     - UAT (user acceptance test)
   - Deployment
   - Documentation
     - Design
     - Code
     - User manual

- Security

8.6 Open Authentication (OAuth)

OAuth is an open standard for access delegation, commonly used as a way for Internet users to grant websites or applications access to their information on other websites but without giving them the passwords.

Source: Wikipedia contributors. (2019, March 19). OAuth. In Wikipedia, The Free Encyclopedia. Retrieved 12:20, March 23, 2019, from https://en.wikipedia.org/w/index.php?title=OAuth&oldid=888559139

8.6.1 Use case for Open Authentication

A user requires access to a resource on a web application (eg StratEx) using her credentials from another website (eg Microsoft Office365).

1. She needs to login using the form from Office365
2. Office365 will generate a token
3. The token is used by StratEx ensuring that the user is effectively logged using her Office365 credentials
4. StratEx can use the resources made available by Office365 such as username, firstname, lastname, email address, read access to OneDrive, write and send new emails. . .

8.6.2 Open authentication using Office365

Microsoft graph documentation makes available Office365 resources of each registrered user:

- https://developer.microsoft.com/en-us/graph
- https://docs.microsoft.com/en-us/graph/overview

Request an access token to Office 365:

- https://login.microsoftonline.com/common/oauth2/v2.0/authorize?client_id=f5d835b0-4bc1-98e7-f98cb4aaef31&scope=https%3A%2F%2Fgraph.microsoft.com%2Fuser.read&response_type=code&redirect_uri=https%3A%2F%2Ftimesheet-stg-inlsprl.azurewebsites.net%2Fsignin-microsoft&state=Ao8m01yi1E76wQIXPJW-F92Fq1v

Description of each parameter (see use-the-authorization-code-to-request-an-access-token[13]):

- https://login.microsoftonline.com/common/oauth2/v2.0/**authorize**
- ? **client_id** =f5d835b0-4bc1-98e7-f98cb4aaef31
- & **scope** =https%3A%2F%2Fgraph.microsoft.com%2Fuser.read
- & **response_type** =code
- & **redirect_uri** =https%3A%2F%2Ftimesheet-stg-inlsprl.azurewebsites.net%2Fsignin-microsoft
- & **state** =Ao8m01yi1E76wQIXPJW-F92Fq1v

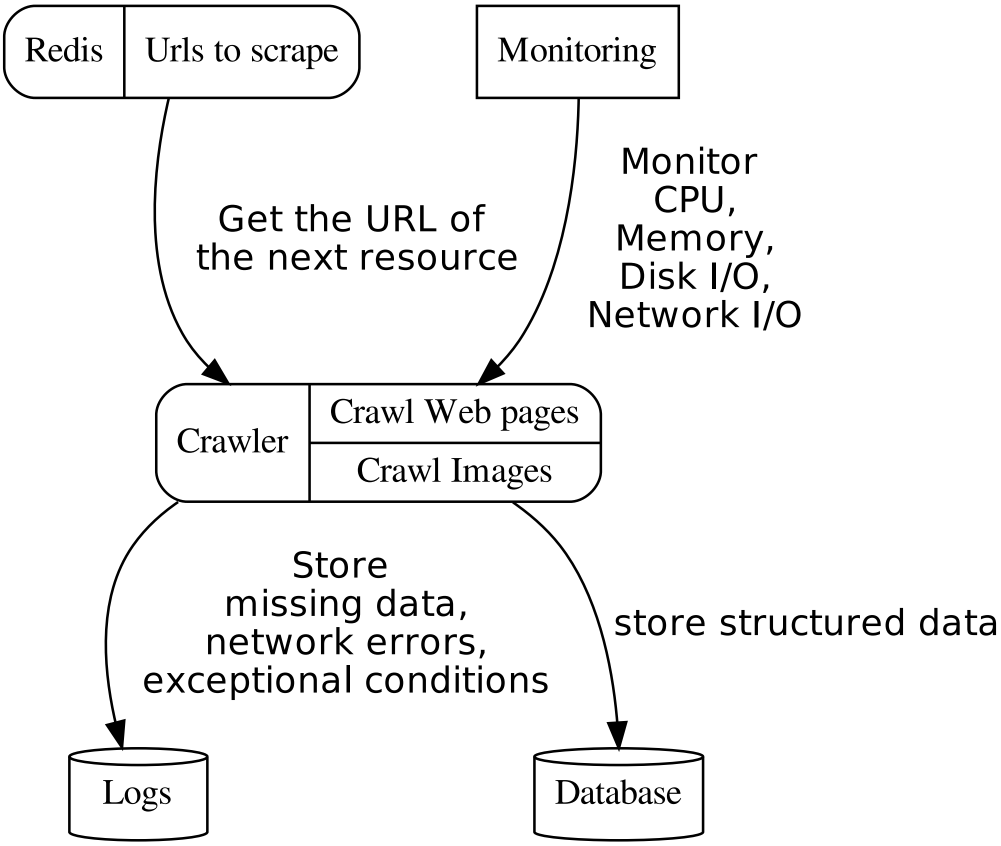
8.7 Web Scraping

8.7.1 Documentation

- https://delftswa.gitbooks.io/desosa-2017/content/scrapy/chapter.html, Joren Hammudoglu (@jorenham), Johan Jonasson (@jojona), Marnix de Graaf (@Eauwzeauw)

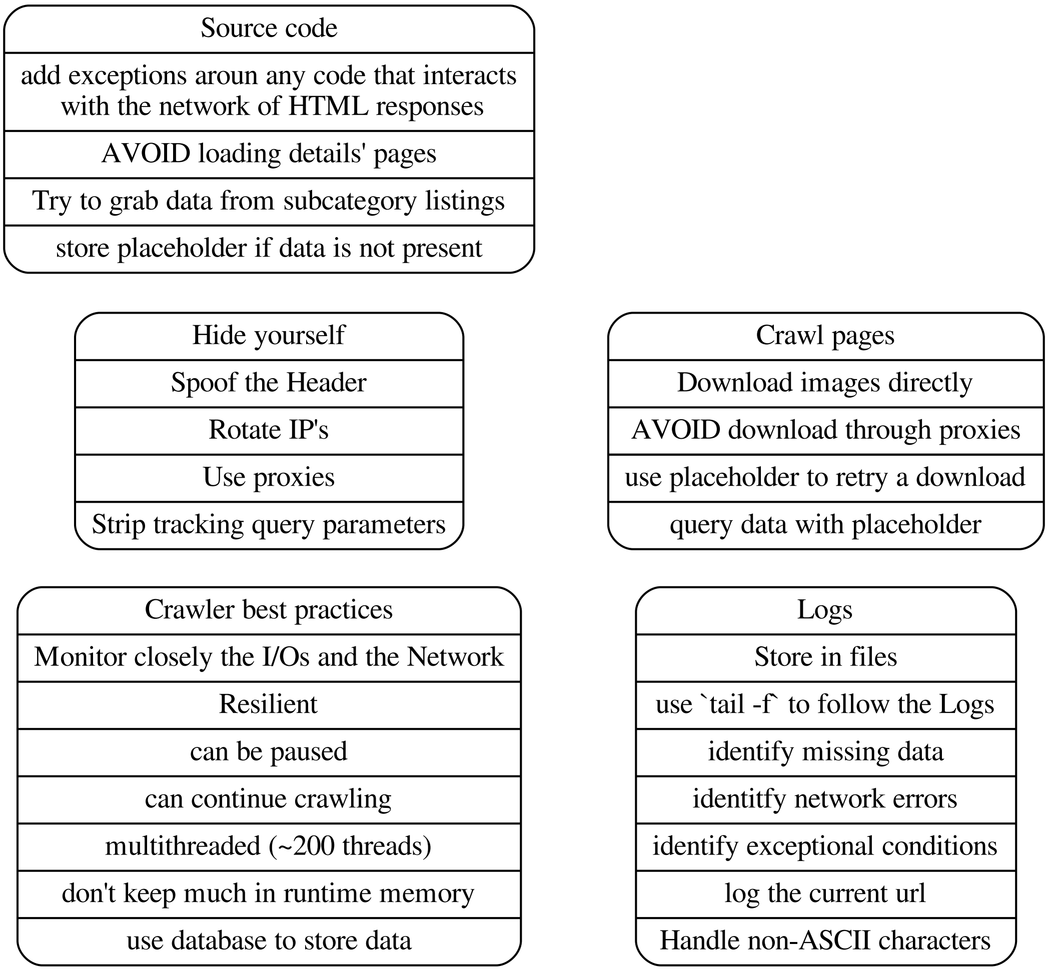
[13] https://docs.microsoft.com/en-us/azure/active-directory/develop/v1-protocols-oauth-code#use-the-authorization-code-to-request-an-access-token

8.7.2 Scraping: Software architecture


Redis
Urls to scrape
Monitoring
Get the URL of the next resource
Monitor CPU, Memory, Disk I/O, Network I/O
Crawler
Crawl Web pages
Crawl Images
Store missing data, network errors, exceptional conditions
store structured data
Logs
Database

Software architecture

8.7.3 Scraping: best practices



Source code
- add exceptions aroun any code that interacts with the network of HTML responses
- AVOID loading details' pages
- Try to grab data from subcategory listings
- store placeholder if data is not present

Hide yourself
- Spoof the Header
- Rotate IP's
- Use proxies
- Strip tracking query parameters

Crawl pages
- Download images directly
- AVOID download through proxies
- use placeholder to retry a download
- query data with placeholder

Crawler best practices
- Monitor closely the I/Os and the Network
- Resilient
- can be paused
- can continue crawling
- multithreaded (~200 threads)
- don't keep much in runtime memory
- use database to store data

Logs
- Store in files
- use `tail -f` to follow the Logs
- identify missing data
- identitfy network errors
- identify exceptional conditions
- log the current url
- Handle non-ASCII characters

Scraping : best practices : Data model v2020-12-26

8.7.4 Extraordinary examples

How To Scrape Amazon Product Data and Prices using Python 3

Source : https://www.scrapehero.com/tutorial-how-to-scrape-amazon-product-details-using-python-and-selectorlib/

selectors.yml

```yaml
name:
    css: '#productTitle'
    type: Text
price:
    css: '#price_inside_buybox'
    type: Text
short_description:
    css: '#featurebullets_feature_div'
```

(continues on next page)

(continued from previous page)

```
    type: Text
images:
    css: '.imgTagWrapper img'
    type: Attribute
    attribute: data-a-dynamic-image
rating:
    css: span.arp-rating-out-of-text
    type: Text
number_of_reviews:
    css: 'a.a-link-normal h2'
    type: Text
variants:
    css: 'form.a-section li'
    multiple: true
    type: Text
    children:
        name:
            css: ""
            type: Attribute
            attribute: title
        asin:
            css: ""
            type: Attribute
            attribute: data-defaultasin
product_description:
    css: '#productDescription'
    type: Text
sales_rank:
    css: 'li#SalesRank'
    type: Text
link_to_all_reviews:
    css: 'div.card-padding a.a-link-emphasis'
    type: Link
```

Amazon.py

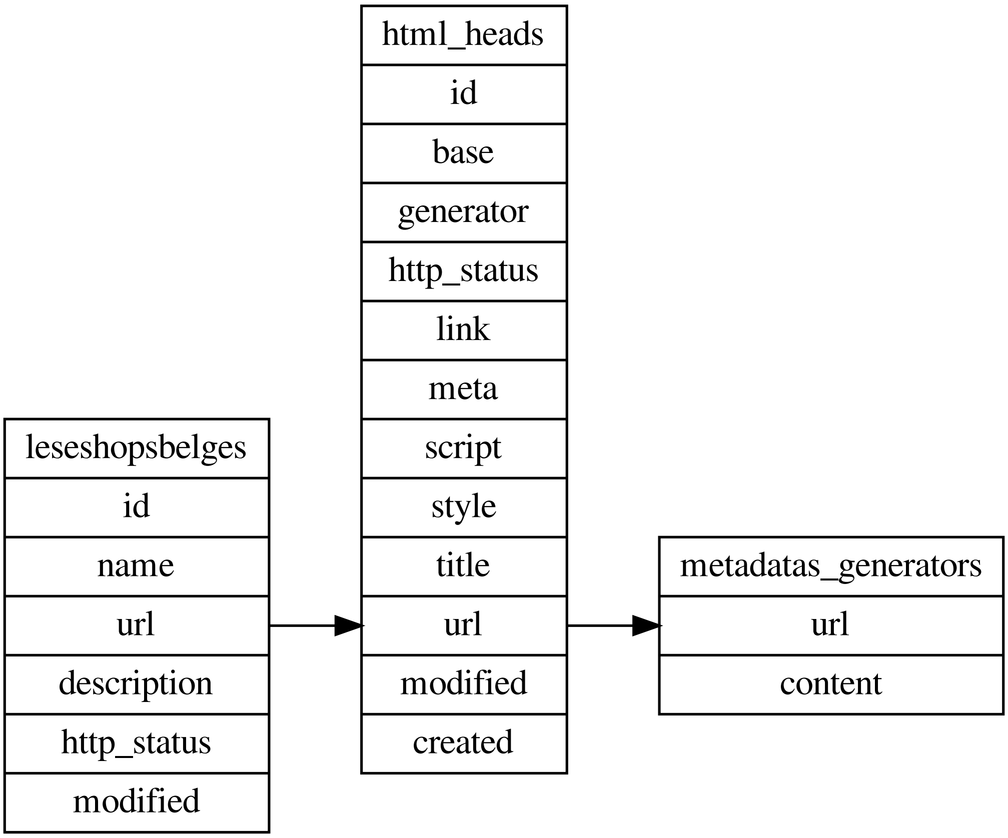
```
from selectorlib import Extractor
import requests
import json
from time import sleep


# Create an Extractor by reading from the YAML file
e = Extractor.from_yaml_file('selectors.yml')

def scrape(url):
    headers = {
        'authority': 'www.amazon.com',
        'pragma': 'no-cache',
        'cache-control': 'no-cache',
        'dnt': '1',
        'upgrade-insecure-requests': '1',
        'user-agent': 'Mozilla/5.0 (X11; CrOS x86_64 8172.45.0) AppleWebKit/537.36␣
↪(KHTML, like Gecko) Chrome/51.0.2704.64 Safari/537.36',
        'accept': 'text/html,application/xhtml+xml,application/xml;q=0.9,image/
↪webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9',
        'sec-fetch-site': 'none',
        'sec-fetch-mode': 'navigate',
        'sec-fetch-dest': 'document',
```

(continues on next page)

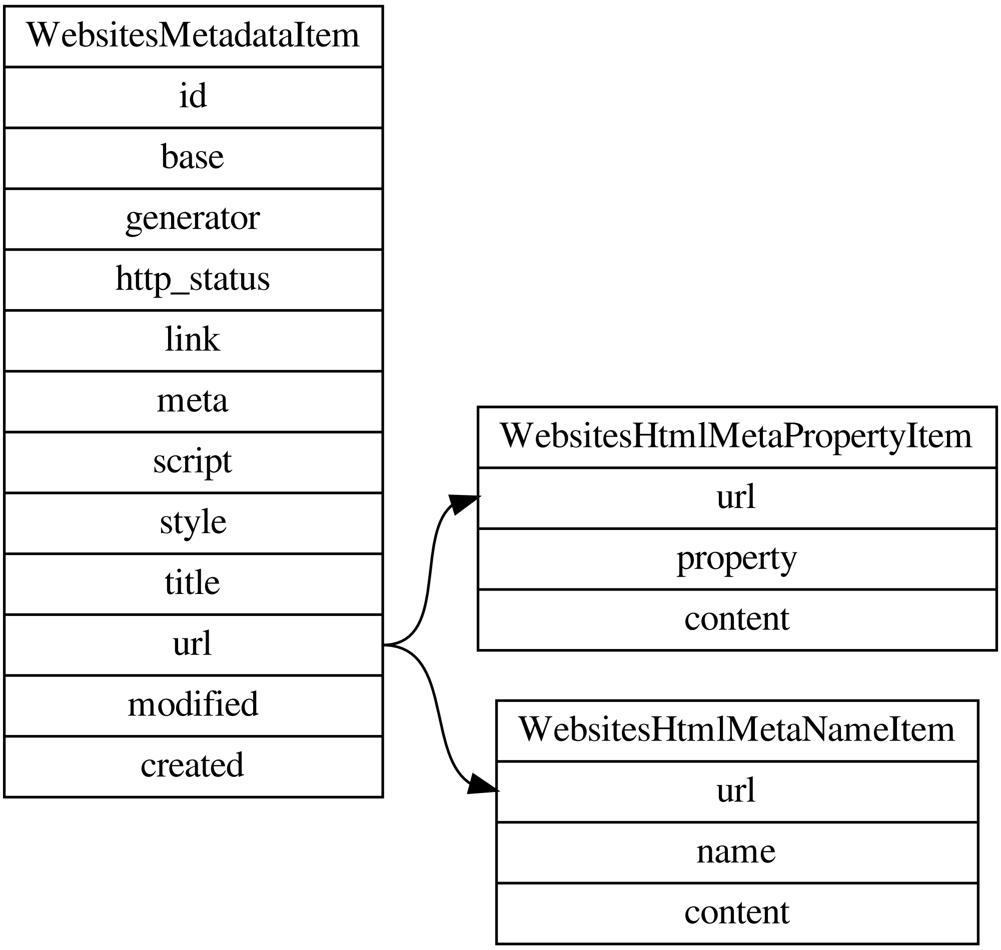
(continued from previous page)

```
        'accept-language': 'en-GB,en-US;q=0.9,en;q=0.8',
    }

    # Download the page using requests
    print("Downloading %s"%url)
    r = requests.get(url, headers=headers)
    # Simple check to check if page was blocked (Usually 503)
    if r.status_code > 500:
        if "To discuss automated access to Amazon data please contact" in r.text:
            print("Page %s was blocked by Amazon. Please try using better proxies\n
↪"%url)
        else:
            print("Page %s must have been blocked by Amazon as the status code was
↪%d"%(url,r.status_code))
        return None
    # Pass the HTML of the page and create
    return e.extract(r.text)

# product_data = []
with open("urls.txt",'r') as urllist, open('output.jsonl','w') as outfile:
    for url in urllist.readlines():
        data = scrape(url)
        if data:
            json.dump(data,outfile)
            outfile.write("\n")
            # sleep(5)
```

8.7.5 Data model of a scraper

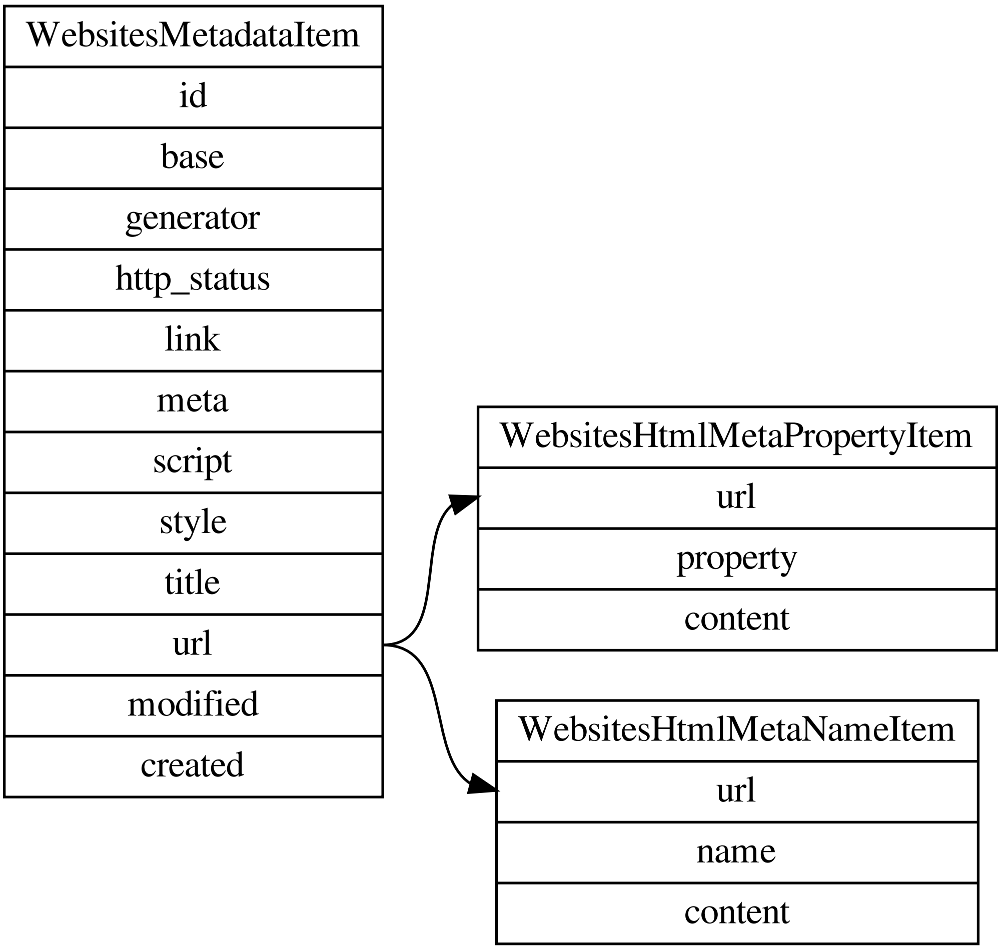


Les eshops belges : Data model v2020-12-26



Scrapy spiders: Items, Data model v2020-12-27

8.7.6 Scraping: Scrapy spiders or Crawlers



Scrapy spiders: Items, Data model v2020-12-27

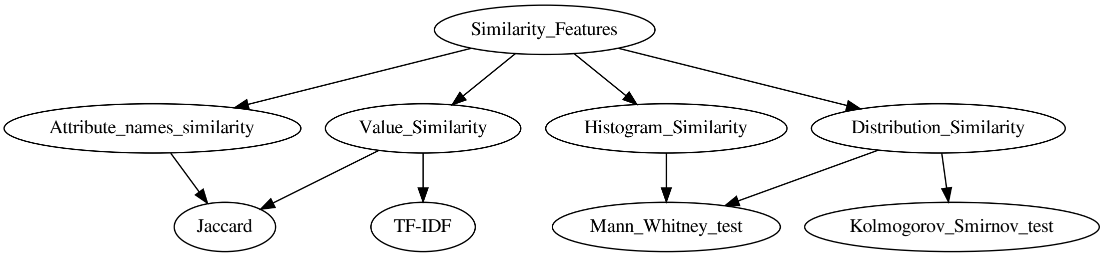
8.7.7 Similarity features

Jaccard: https://en.wikipedia.org/wiki/Jaccard_index

TD-IDF, term frequency–inverse document frequency: https://en.wikipedia.org/wiki/Tf%E2%80%93idf

Mann–Whitney U test is a nonparametric test of the null hypothesis that, for randomly selected values X and Y from two populations, the probability of X being greater than Y is equal to the probability of Y being greater than X.

Kolmogorov–Smirnov test In statistics, the Kolmogorov–Smirnov test (K–S test or KS test) is a nonparametric test of the equality of continuous (or discontinuous, see Section 2.2), one-dimensional probability distributions that can be used to compare a sample with a reference probability distribution (one-sample K–S test), or to compare two samples (two-sample K–S test).



Similarity features v2020-12-23. Source: 2018-02-03-AAAI-KG-Tutorial-CK.pptx

Source : AAAI 2018 Tutorial Building Knowledge Graphs

8.8 Authentic sources

- Een authentieke bron is een gegevensbank waarin authentieke gegevens worden bewaard en die geldt als dé referentie voor deze specifieke gegevens over personen en rechtsfeiten. Bron: https://dt.bosa.be/nl/authentieke_bronnen
- Une source authentique est une banque de données au sein de laquelle des données authentiques sont conservées et qui sert de référence pour ces données spécifiques sur des personnes et faits juridiques. Source: https://dt.bosa.be/fr/echange_de_donnees/sources_authentiques

8.8.1 Geography - Countries

- Get information about countries via a RESTful API : https://restcountries.eu

9 HR management (Curriculum)

Human Resource Management - the activities associated with recruiting,development, and compensation of employees.

9.1 Some rules to help you run your company

- Remove rules
- Be candid with each others
- Hire adults
- Hire best people you can
- Give freedom
- Give responsibility

9.2 .NET Curriculum

9.2.1 Documentation

- sphinx : https://www.sphinx-doc.org/en/master/usage/restructuredtext/basics.html
    1. install sphinx
    2. install template rtfd

Todo:

3. create a public repo on github presenting a template for sphinx projects

---

9.2.2 Testing

- Creating Unit Tests for ASP.NET MVC Applications (C#) :
    - https://docs.microsoft.com/en-us/aspnet/mvc/overview/older-versions-1/unit-testing/creating-unit-tests-for-asp-net-mvc-applications-cs
- Testing and debuggin ASP.NET Web API :
    - Testing and Debugging ASP.NET Web API[14]
- SeleniumHQ Browser automation :
    - https://www.seleniumhq.org
- End to end testing angular :
    - https://www.protractortest.org

9.2.3 Software engineering

- The Joel Test - 12 Steps to Better Code[15]
- The Joel Test For Programmers (The Simple Programmer Test)[16]

9.2.4 Software development .NET

- Microsoft .NET[17] : C#
- Debugging code back-end (Visual Studio)
- Debugging code front-end (Browser, Visual Studio)
- Debugging code on Azure (Browser, Visual Studio)
- MVC : ASP.NET MVC[18] Model-View-Controller
- CSS : https://developer.mozilla.org/en-US/docs/Web/CSS
- Bootstrap 4 : https://getbootstrap.com/
- Authentication/Authorization: .net identity[19]

[14] https://docs.microsoft.com/en-us/aspnet/web-api/overview/testing-and-debugging
[15] https://www.joelonsoftware.com/2000/08/09/the-joel-test-12-steps-to-better-code
[16] https://simpleprogrammer.com/joel-test-programmers-simple-programmer-test
[17] https://www.microsoft.com/net
[18] https://www.asp.net/mvc
[19] https://docs.microsoft.com/en-us/aspnet/core/security/authentication/identity?view=aspnetcore-2.1&tabs=visual-studio

- REST API: securization by token containing a UUID + Claims. Log using MS Account -> Get VSTS tokens -> display Personal projects
- REST API (documentation/deployment): SWAGGER[20]
- REST API (test) : POSTMAN[21]
- SPA Single Page Application
  - ANGULAR
    - Linter[22] : https://angular.io/cli/lint
  - Typescript
    - Linter[23] : https://github.com/Microsoft/dtslint
- NUGET: back-end code + database + migration database + INL/Metis repository

9.2.5 DATA MODELING .NET and Azure

- Entity framework[24]: Code first
- Database:
  - SQL (SQL Server), Azure SQL Server[25] classic (do not use elastic pool)
  - NoSQL: CosmosDB[26]
- File storage[27] (SMB 3.0)
- Media (photo, video, sound): Azure Media services[28]

9.2.6 GENERATE OFFICE FILES

- File Format APIs : https://www.aspose.com

9.2.7 DEPLOYMENT

- Team Services: Understand the agile methodology[29]
- Azure Web App : https://portal.azure.com
- Azure Pipelines : https://azure.microsoft.com/en-us/services/devops/pipelines
- Jenkins supports building, deploying and automating any project.: https://jenkins.io/

[20] https://swagger.io/
[21] https://www.getpostman.com/
[22] https://en.wikipedia.org/wiki/Lint_(software)
[23] https://en.wikipedia.org/wiki/Lint_(software)
[24] https://docs.microsoft.com/en-us/ef/
[25] https://azure.microsoft.com/en-us/services/sql-database/
[26] https://docs.microsoft.com/en-us/azure/cosmos-db/
[27] https://azure.microsoft.com/en-us/services/storage/files/
[28] https://azure.microsoft.com/en-us/services/media-services/
[29] https://www.visualstudio.com/learn/scale-agile-large-teams

9.2.8 Artificial intelligence / Machine learning / Big data analysis

- Artificial intelligence: https://azure.microsoft.com/en-us/services/cognitive-services
- Machine learning: https://azure.microsoft.com/en-us/services/machine-learning-services
- NTLK Natural Language Toolkit : https://www.nltk.org

9.2.9 Reproducible research

- Tools
  - Jupyter notebook are documents produced by the Jupyter Notebook App, which contain both computer code (e.g. python) and rich text elements (paragraph, equations, figures, links, etc): http://jupyter.org
  - nteract and create with data, words, and visuals : https://nteract.io/
  - Microsoft Azure Notebooks : http://notebooks.azure.com
  - Colabotary : https://colab.research.google.com/
- Theory
  - Reproducable research: https://mg.readthedocs.io/reproducible_research.html
  - Reproducibility Workshop: Best practices and easy steps to save time for yourself and other researchers: https://codeocean.com/workshop/caltech

9.3 Excel Curriculum

9.3.1 Lookups

- Introduction
- VLOOKUP
- VLOOKUP Exact Match
- HLOOKUP
- HLOOKUP Exact Match

9.3.2 Conditional Logic

- Introduction
- IF Statement
- Nested IF
- AND
- OR
- NOT
- IFERROR
- SUMIF
- AVERAGEIF
- COUNTIF & COUNTIF
- SUMIFS
- AVERAGEIFS

9.3.3 Data Tools

- Introduction
- Data Validation
- Drop-Down Lists
- Removing Duplicates
- Text To Columns
- Goal Seek
- Scenario Manager

9.3.4 PivotTables

- Introduction
- Creating PivotTables
- Choosing Fields
- PivotTable Layout
- Filtering PivotTables
- Modifying PivotTable Data
- PivotCharts

9.3.5 Collaboration

- Introduction
- Document Properties
- Inserting Hyperlinks
- Sharing a Workbook
- Track Changes
- Accept/Reject Changes
- Mark as Final

9.3.6 Printing

- Introduction
- Page Orientation
- Page Breaks
- Print Area
- Margins
- Print Titles
- Headers and Footers
- Scaling
- Sheet Options

9.3.7 Macros

- Introduction and Macro Security
- Recording a Macro
- Assign a Macro to a Button or Shape
- Run a Macro upon Opening a Workbook
- How to Inspect and Modify a Macro

9.4 Secure and Private Artificial Intelligence

- course source :
  - https://classroom.udacity.com/courses/ud185
  - other course : https://www.udacity.com/school-of-ai

9.4.1 Deep learning with PyTorch

- https://research.fb.com/category/facebook-ai-research
- Notebooks : https://github.com/udacity/deep-learning-v2-pytorch

Install Python3

- create a python3.7.X environment : *conda create -n py37 python=3.7 anaconda*
- activate the environment *conda activate py37*
- deactivate the environment *conda deactivate*
- determining my environment : *conda info –envs*

Install PyTorch an Conda

- Install Conda
  - install Anaconda : https://docs.anaconda.com/anaconda/install and https://conda.io/en/latest
  - or install Miniconda : https://docs.conda.io/en/latest/miniconda.html
  - Some commands
    - managing environments : https://conda.io/projects/conda/en/latest/user-guide/getting-started.html#managing-environments
    - example of commands : *conda search scipy*, *conda install scipy*, *conda build my_fun_package*, *conda update conda*
- Install PyTorch https://pytorch.org/get-started/locally
  - for old GPU (does not work on Geforce GT 520M *conda install pytorch torchvision cudatoolkit=9.0 -c pytorch -c defaults -c numba/label/dev*
  - *conda install pytorch torchvision cudatoolkit=10.0 -c pytorch*
- Install numpy, jupyter and notebook
  - *conda install numpy jupyter notebook*

Launching Jupyter Notebook App

- *jupyter notebook* - https://jupyter-notebook-beginner-guide.readthedocs.io/en/latest/execute.html

9.4.2 Udacity course : Deep Learning with PyTorch

This repo contains notebooks and related code for Udacity's Deep Learning with PyTorch lesson. This lesson appears in our [AI Programming with Python Nanodegree program](https://www.udacity.com/course/ai-programming-python-nanodegree–nd089[30]).

- **Part 1:** Introduction to PyTorch and using tensors
- **Part 2:** Building fully-connected neural networks with PyTorch
- **Part 3:** How to train a fully-connected network with backpropagation on MNIST
- **Part 4:** Exercise - train a neural network on Fashion-MNIST
- **Part 5:** Using a trained network for making predictions and validating networks
- **Part 6:** How to save and load trained models
- **Part 7:** Load image data with torchvision, also data augmentation
- **Part 8:** Use transfer learning to train a state-of-the-art image classifier for dogs and cats

9.4.3 Tools for Artificial Intelligence

- Gym is a toolkit for developing and comparing reinforcement learning algorithms. It supports teaching agents everything from walking to playing games like Pong or Pinball. https://gym.openai.com
- ONNX is an open format to represent deep learning models. With ONNX, AI developers can more easily move models between state-of-the-art tools and choose the combination that is best for them. ONNX is developed and supported by a community of partners. https://onnx.ai
- Machine learning cheatsheet : https://ml-cheatsheet.readthedocs.io

10 Firm infrastructure (copyrights, CISSP, GDPR)

Firm Infrastructure - includes activities such as finance, legal, quality management, etc.

10.1 Copyrights

- Choose a creative commons license : https://creativecommons.org/choose/
- Software Licenses in Plain English : https://tldrlegal.com/

[30] https://www.udacity.com/course/ai-programming-python-nanodegree--nd089

10.2 CISSP

10.2.1 Business Continuity Plan

When business is disrupted, it can cost money. Lost revenues plus extra expenses means reduced profits. Insurance does not cover all costs and cannot replace customers that defect to the competition. A business continuity plan to continue business is essential. Development of a business continuity plan includes four steps:

- Conduct a business impact analysis to identify time-sensitive or critical business functions and processes and the resources that support them.
- Identify, document, and implement to recover critical business functions and processes.
- Organize a business continuity team and compile a business continuity plan to manage a business disruption.
- Conduct training for the business continuity team and testing and exercises to evaluate recovery strategies and the plan.
- Source Official website of the Department of Homeland Security : https://www.ready.gov/business/implementation/continuity

10.2.2 Disaster Recovery Plan

An information technology disaster recovery plan (IT DRP) should be developed in conjunction with the business continuity plan. Priorities and recovery time objectives for information technology should be developed during the business impact analysis. Technology recovery strategies should be developed to restore hardware, applications and data in time to meet the needs of the business recovery.

Source Official website of the Department of Homeland Security : https://www.ready.gov/business/implementation/IT

10.3 GDPR - Règlement général sur la protection des données

Le règlement no 2016/679, dit règlement général sur la protection des données (RGPD, ou encore GDPR, de l'anglais General Data Protection Regulation), est un règlement de l'Union européenne qui constitue le texte de référence en matière de protection des données à caractère personnel1. Il renforce et unifie la protection des données pour les individus au sein de l'Union européenne.

Après quatre années de négociations législatives, ce règlement a été définitivement adopté par le Parlement européen le 14 avril 2016. Ses dispositions sont directement applicables dans l'ensemble des 28 États membres de l'Union européenne à compter du 25 mai 2018.

Ce règlement remplace la directive sur la protection des données personnelles adoptée en 1995 (article 94 du règlement) ; contrairement aux directives, les règlements n'impliquent pas que les États membres adoptent une loi de transposition pour être applicables2.

Les principaux objectifs du RGPD sont d'accroître à la fois la protection des personnes concernées par un traitement de leurs données à caractère personnel et la responsabilisation des acteurs de ce traitement. Ces principes pourront être appliqués grâce à l'augmentation du pouvoir des autorités de régulation.

- European Commission : https://ec.europa.eu/info/law/law-topic/data-protection_en
- Wikipedia : https://en.wikipedia.org/wiki/General_Data_Protection_Regulation

10.3.1 PROCESS

- Describe the project
- Ensure that a clear consent is described and the user clicks on the checkbox (sentence is in the active voice + describe with precision the kind of treatment)
- Describe the treatments of personal data
- Describe the sub contracts such as Microsoft AppInsight, Google Analytics, Insurance, Bank

10.3.2 DATA MODEL

How to store the consent?

How to build the audit trail? - application name - which operation has been triggered? - who performed the operation - list the dossiers that have been exported (csv format) - Current dateTime - url used to generate the log

LOG THE CONSENT

- application name - user who consent - date of the consent- version of the consent - the text of the consent

10.4 Legal Regulations Compliance

- FSMA : https://www.fsma.be/fr
- GDPR : https://ec.europa.eu/info/law/law-topic/data-protection_en
- PSD2 : https://ec.europa.eu/info/law/payment-services-psd-2-directive-eu-2015-2366_en

11 Misc.

11.1 Technologies

11.1.1 www.stoic.com

http://www.stoic.com (tool generating applications from a spreadsheet) - but as you have said, this won’t work at EADS because they don’t allow javascritps to run on their browsers Technologies used by stoic: https://bitbucket.org/altf1be/software-architecture/wiki/Technologies

11.1.2 how to stoic.com generate an application from a spreadsheet?

When I have met the founder of stoic last week he told me they generate the applications in this way:

1. they import the spreadsheet and they try to automatically recognize the fields (date, name, address, etc.)
2. they store the data into 3 different databases:
   1. http://www.postgresql.org/ - relational database to do basic queries
   2. http://www.mongodb.org/ - nosql database storing all the documents
   3. http://www.elasticsearch.org/ - to search into the 2 previous databases
3. They give the means (widgets) for business users to create rules, user interfaces and perspectives (different views of the same application e.g. calendar view, google maps view. . . .)
4. they run the application on www.nodejs.org
5. They deploy the application on http://www.cloudfoundry.com/

11.1.3 stoic.com business model

Their business model is based on SaaS subscriptions; they plan to deliver a package one could deploy on its own servers.

11.1.4 stoic.com competitive advantage

Stoic founder claims that he can generate an application like a sales force automation or www.stackoverflow.com applications within 3-4 days with business experts

11.1.5 stoic.com uses those Javascript technologies

src : http://stoic.com/stack/#/questions/446

- (http://josscrowcroft.github.com/accounting.js/) accounting.js (currency formatting)
- (http://segmentio.github.com/analytics.js/) Analytics.js (web analytics)
- (http://angularjs.org/) AngularJS (user interface framework)
- (http://angular-ui.github.com/) AngularUI (user interface components)
- (http://jhollingworth.github.com/bootstrap-wysihtml5/) bootstrap-wysihtml5 (rich text editor)
- (https://github.com/substack/node-browserify) Browserify (dependency manager)
- (http://hughsk.github.com/colony/) Colony (graph visualization)
- (http://www.eyecon.ro/bootstrap-colorpicker/) Colorpicker for Bootstrap (color picker)
- (http://www.senchalabs.org/connect/) Connect (middleware framework)
- (http://www.jacksasylum.eu/ContentFlow/) ContentFlow (coverflow)
- (http://code.google.com/p/cookie-js/) Cookie.js (cookie library)
- (http://d3js.org/) D3 (visualization library)
- (https://github.com/visionmedia/debug/) debug (logger)
- (https://github.com/eleith/emailjs) emailjs (SMTP client)
- (https://github.com/bnoguchi/everyauth/) everyauth (authentication and authorization package)
- (http://bgrins.github.com/ExpandingTextareas/) Expanding Textareas (expanding textareas)
- (http://expressjs.com/) express (web application framework)
- (https://github.com/cloudhead/eyes.js/tree/) eyes (value inspector)
- (http://fortawesome.github.com/Font-Awesome/) Font Awesome (icon font)
- (http://stoic.com/formula/) Formula.js (formula functions)
- (http://arshaw.com/fullcalendar/) FullCalendar (calendar)
- (https://github.com/jquery/globalize/) Globalize (globalization and localization library)
- (https://github.com/jeff-optimizely/Guiders-JS) Guiders.js (on-page guidance)
- (http://gotwarlost.github.com/istanbul/) istanbul (code coverage tool)
- (http://jade-lang.com/) Jade (template engine)
- (http://silentmatt.com/javascript-expression-evaluator/) JavaScript Expression Evaluator (expression evaluator)
- (http://slexaxton.github.com/Jed/) Jed (internationalization library)
- (http://www.jplayer.org/) jPlayer (media player)

- (http://jquery.com/) jQuery (HTML library)
- (http://masonry.desandro.com/) jQuery Masonry (dynamic layout)
- (http://jquerymobile.com/) jQuery Mobile (mobile user interface library)
- (http://omnipotent.net/jquery.sparkline/) jQuery Sparklines (sparklines)
- (http://jqueryui.com/) jQuery UI (user interface library)
- (http://miller.mageekbox.net/) jQuery.miller.js (miller columns)
- (https://github.com/sgruhier/jquery-addresspicker) jquery-addresspicker (address picker)
- (http://code.google.com/p/jquery-ui-map/) jquery-ui-map (maps)
- (http://jsdox.org) jsdox (API documentation generator)
- (https://github.com/jshint/jshint/) JSHint (code analysis tool)
- (http://jsoneditoronline.org/) JSON Editor Online (JSON editor)
- (http://jstat.org/) jStat (statistics library)
- (http://learnboost.github.com/kue/) Kue (job queue)
- (http://lesscss.org/) LESS (dynamic stylesheet compiler)
- (http://lodash.com/) Lo-Dash (functional programming library)
- (http://github.com/sutoiku/mapperjs/) mapper.js (object data mapper)
- (http://github.com/chjj/marked) marked (markdown compiler)
- (http://digitalbush.com/projects/masked-input-plugin/) Masked Input (input mask library)
- (http://masonry.desandro.com/) Masonry (dynamic layout)
- (http://www.mathjax.org/) MathJax (mathematics display engine)
- (https://github.com/mikejihbe/metrics) Metrics (metrics library)
- (http://visionmedia.github.com/mocha/) mocha (test framework)
- (http://modernizr.com/) Modernizr (feature detection)
- (http://momentjs.com/) Moment.js (date library)
- (http://josscrowcroft.github.com/money.js/) money.js (currency conversion)
- (https://github.com/brianc/node-postgres/) node-postgres (PostgreSQL client)
- (https://github.com/broofa/node-uuid/) node-uuid (UUID generator)
- (https://github.com/C2FO/nools) Nools (business rules engine)
- (http://numeraljs.com/) Numeral.js (number library)
- (http://numericjs.com/) Numeric Javascript (numeric library)
- (http://parseqjs.com) parseq.js (control-flow library)
- (http://phantomjs.org/) PhantomJS (headless WebKit)
- (http://code.shutterstock.com/rickshaw/) Rickshaw (time series graphs)
- (https://github.com/visionmedia/should.js/tree/) should (assertion library)
- (https://github.com/mleibman/SlickGrid) SlickGrid (grid)
- (http://socket.io/) Socket.IO (socket library)
- (https://github.com/fschaefer/Stately.js) Stately.js (finite state machine)
- (http://vojtajina.github.com/testacular/) Testacular (test runner)
- (http://timeline.verite.co/) TimelineJS (timeline)

- (http://twitter.github.com/bootstrap/) Twitter Bootstrap (HTML5 toolkit)
- (https://github.com/mishoo/UglifyJS/) UglifyJS (parser, compressor, beautifier)
- (http://epeli.github.com/underscore.string/) Underscore.string (string library)
- (http://vexflow.com/) VexFlow (music notation rendering)
- (https://github.com/ryanmcgrath/wrench-js) wrench.js (recursive file operations library)

12 Glossary

Activation function An activation function calculates a "weighted sum" of its input, adds a bias and then decides whether it should be "fired" or not

See Activation Functions[31]

See Understanding Activation Functions in Neural Network[32] $y = f(\sum_i wixi + b)$

alt-f1 ALT-F1 designs, implements, deploys and supports secure, large-scale software solutions for diverse industries: Manufacturing, MRO, Warehouse, Broadcasting, Bank, Insurance, Law Enforcement, Justice & Serious International Crime

See http://www.alt-f1.be

autograd Module that PyTorch uses to calculate gradients for training neural networks

See https://pytorch.org/docs/stable/notes/autograd.html

Back office *The back office is all the resources of the company that are devoted to actually producing a product or service and all the other labor that isn't seen by customers, such as administration or logistics.*

Source: Wikipedia contributors. (2019, July 19). Back office. In Wikipedia, The Free Encyclopedia. Retrieved 07:53, September 19, 2019, from https://en.wikipedia.org/w/index.php?title=Back_office&oldid=906961159

Broker *"An insurance broker sells, solicits, or negotiates insurance for compensation."*

Source: Wikipedia contributors. (2019, September 12). Insurance broker. In Wikipedia, The Free Encyclopedia. Retrieved 10:33, September 13, 2019, from https://en.wikipedia.org/w/index.php?title=Insurance_broker&oldid=915277342

Business Process Management Business Process Management *is a discipline aimed at managing all aspect of the business processes; from process design to modeling and analysis to execution and improvement*

Source: https://www.ipdsolution.com/ipdblog/bpm-workflows

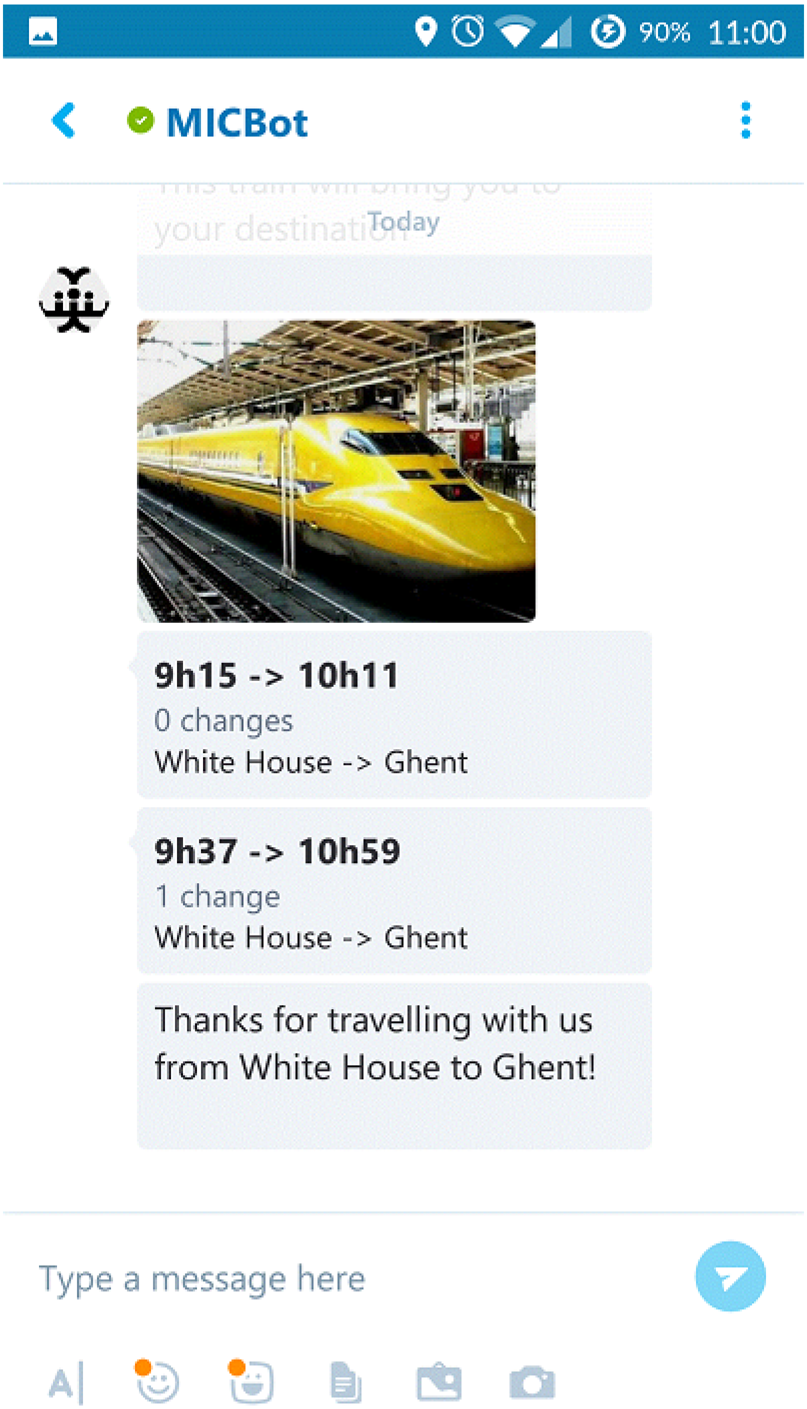
CDI See term:*Customer Data Integration*

Chatbot *"A chatbot is a piece of software that conducts a conversation via auditory or textual methods."*

Source: Wikipedia contributors. (2019, September 9). Chatbot. In Wikipedia, The Free Encyclopedia. Retrieved 14:26, September 12, 2019, from https://en.wikipedia.org/w/index.php?title=Chatbot&oldid=914875664

[31] https://ml-cheatsheet.readthedocs.io/en/latest/activation_functions.html

[32] https://medium.com/the-theory-of-everything/understanding-activation-functions-in-neural-networks-9491262884e0



9h15 -> 10h11
0 changes
White House -> Ghent

9h37 -> 10h59
1 change
White House -> Ghent

Thanks for travelling with us from White House to Ghent!

Type a message here

Conda Package, dependency and environment management for any language—Python, R, Ruby, Lua, Scala, Java, JavaScript, C/ C++, FORTRAN

Contact center *A contact center, further extension to call centers administers centralized handling of individual communications, including letters, faxes, live support software, social media, instant message, and e-mail.*

Source: Wikipedia contributors. (2019, September 15). Call centre. In Wikipedia, The Free Encyclopedia. Retrieved 08:59, September 19, 2019, from https://en.wikipedia.org/w/index.php?title=Call_centre&oldid=915792349

Cross-entropy loss Cross-entropy loss, or log loss, measures the performance of a classification model whose output is a probability value between 0 and 1.

A perfect model would have a log loss of 0.

See Neural networks - Cross Entropy[33]

See PyTorch - Cross entropy loss function[34]

CUDA PyTorch uses a library called CUDA to accelerate operations using the GPU

Customer Data Integration *"Customer data integration (CDI) is the process of defining, consolidating and managing customer information across an organization's business units and systems to achieve a "single version of the truth" for customer data."*

Source: https://searchdatamanagement.techtarget.com/definition/customer-data-integration

Digitization Digitization is *"Digitization, less commonly digitalization, is the process of converting information into a digital (i.e. computer-readable) format, in which the information is organized into bits."*

Source: Wikipedia contributors. (2019, August 28). Digitization. In Wikipedia, The Free Encyclopedia. Retrieved 07:13, September 12, 2019, from https://en.wikipedia.org/w/index.php?title=Digitization&oldid=912864588

Epoch One Epoch is when an ENTIRE dataset is passed forward and backward through the neural network only ONCE.

See Epoch vs Batch Size vs Iterations[35]

Gradient descent The gradient is the slope of the loss function and points in the direction of fastest change. To get to the minimum in the least amount of time, we then want to follow the gradient (downwards). You can think of this like descending a mountain by following the steepest slope to the base.

See Intro to PyTorch - Notebook Workspace[36]

Gradients Gradient descent is an optimization algorithm used to minimize some function by iteratively moving in the direction of steepest descent as defined by the negative of the gradient.

In machine learning, we use *Gradient descent* to update the parameters of our model. Parameters refer to coefficients in Linear Regression and weights in neural networks.

See https://ml-cheatsheet.readthedocs.io/en/latest/gradient_descent.html

A gradient is a partial derivative - why partial? Because one computes it with respect to (w.r.t.) a single parameter. We have two parameters, a and b, so we must compute two partial derivatives

See Understanding PyTorch with an example: a step-by-step tutorial[37]

Hidden Layers Sits between the input and output layers and applies an activation function before passing on the results.

There are often multiple hidden layers in a network.

In traditional networks, hidden layers are typically fully-connected layers - each neuron receives input from all the previous layer's neurons and sends its output to every neuron in the next layer.

See https://ml-cheatsheet.readthedocs.io/en/latest/nn_concepts.html?highlight=hidden#layers

Inbound call center *An inbound call center is operated by a company to administer incoming product or service support or information enquiries from consumers.*

Source: Wikipedia contributors. (2019, September 15). Call centre. In Wikipedia, The Free Encyclopedia. Retrieved 08:59, September 19, 2019, from https://en.wikipedia.org/w/index.php?title=Call_centre&oldid=915792349

Jaccard *The Jaccard index, also known as Intersection over Union and the Jaccard similarity coefficient (originally given the French name coefficient de communauté by Paul Jaccard), is a statistic used for gauging the similarity and diversity of sample sets. The Jaccard coefficient measures similarity between finite sample sets, and is defined as the size of the intersection divided by the size of the union of the sample sets*

[33] https://ml-cheatsheet.readthedocs.io/en/latest/loss_functions.html?highlight=cross-entropy%20loss%20#cross-entropy

[34] https://pytorch.org/docs/stable/nn.html#torch.nn.CrossEntropyLoss

[35] https://towardsdatascience.com/epoch-vs-iterations-vs-batch-size-4dfb9c7ce9c9

[36] https://classroom.udacity.com/courses/ud185/lessons/8a993162-65c4-4a80-bd35-47d9f3a6f5bc/concepts/70526adf-40d3-4446-ac32-d3f798739745

[37] https://towardsdatascience.com/understanding-pytorch-with-an-example-a-step-by-step-tutorial-81fc5f8c4e8e

Source: https://en.wikipedia.org/wiki/Jaccard_index

Kolmogorov–Smirnov test *In statistics, the Kolmogorov–Smirnov test (K–S test or KS test) is a nonparametric test of the equality of continuous (or discontinuous, see Section 2.2), one-dimensional probability distributions that can be used to compare a sample with a reference probability distribution (one-sample K–S test), or to compare two samples (two-sample K–S test).*

Source: https://en.wikipedia.org/wiki/Kolmogorov%E2%80%93Smirnov_test

Layers The first layer shown on the bottom here are the inputs, understandably called the input layer. The middle layer is called the hidden layer, and the final layer (on the right) is the output layer.

Source: Intro to PyTorch - Notebook Workspace[38]

Logit In statistics, the logit function or the log-odds is the logarithm of the odds p/(1 - p) where p is probability. It is a type of function that creates a map of probability values from [0,1] to $-\infty, +\infty$

It is the inverse of the sigmoidal "logistic" function or logistic transform used in mathematics, especially in statistics.

See https://en.wikipedia.org/wiki/Logit

s function

A measure of our prediction error. (also called the cost)

Mann–Whitney U test *In statistics, the Mann–Whitney U test (also called the Mann–Whitney–Wilcoxon (MWW), Wilcoxon rank-sum test, or Wilcoxon–Mann–Whitney test) is a nonparametric test of the null hypothesis that, for randomly selected values X and Y from two populations, the probability of X being greater than Y is equal to the probability of Y being greater than X.*

Source: https://en.wikipedia.org/wiki/Mann%E2%80%93Whitney_U_test

mathjax See Short Math Guide for LATEX[39]

See Math into LATEX An Introduction to LATEX and AMS-LATEX[40]

Middle office *The middle office is made up of the risk managers and the information technology managers who manage risk and maintain the information resources.*

Source: Wikipedia contributors. (2019, August 9). Middle office. In Wikipedia, The Free Encyclopedia. Retrieved 08:36, September 19, 2019, from https://en.wikipedia.org/w/index.php?title=Middle_office&oldid=910135163

MNIST *The Modified National Institute of Standards and Technology database is a large database of handwritten digits that is commonly used for training various image processing systems.* Source https://en.wikipedia.org/wiki/MNIST_database

NumPy Interacts with term:*PyTorch*. NumPy is the fundamental package for scientific computing with Python. It contains among other things:

- a powerful N-dimensional array object
- sophisticated (broadcasting) functions
- tools for integrating C/C++ and Fortran code
- useful linear algebra, Fourier transform, and random number capabilities

See https://numpy.org/

OpenMined OpenMined is an open-source community focused on researching, developing, and promoting tools for secure, privacy-preserving, value-aligned artificial intelligence. https://www.openmined.org

[38] https://classroom.udacity.com/courses/ud185/lessons/8a993162-65c4-4a80-bd35-47d9f3a6f5bc/concepts/70526adf-40d3-4446-ac32-d3f798739745

[39] http://mirror.koddos.net/CTAN/info/short-math-guide/short-math-guide.pdf

[40] http://mirrors.fe.up.pt/pub/CTAN/info/examples/mil/mil.pdf

Outbound call center *An outbound call center is operated for telemarketing, for solicitation of charitable or political donations, debt collection, market research, emergency notifications, and urgent/critical needs blood banks.*

Source: Wikipedia contributors. (2019, September 15). Call centre. In Wikipedia, The Free Encyclopedia. Retrieved 08:59, September 19, 2019, from https://en.wikipedia.org/w/index.php?title=Call_centre&oldid=915792349

PyTorch An open source machine learning framework that accelerates the path from research prototyping to production deployment.

See https://pytorch.org/

Robo advisor *a class of financial adviser that provide financial advice or Investment management online with moderate to minimal human intervention*

Source: Wikipedia contributors. (2019, August 29). Robo-advisor. In Wikipedia, The Free Encyclopedia. Retrieved 14:22, September 12, 2019, from https://en.wikipedia.org/w/index.php?title=Robo-advisor&oldid=912998258

Sigmoid function A sigmoid function is a mathematical function having a characteristic "S"-shaped curve or sigmoid curve.

See https://en.wikipedia.org/wiki/Sigmoid

SIREMIS Web Management Interface for Kamailio (OpenSER) SIP Server

See https://siremis.asipto.com

td-idf

TD-IDF

Term Frequency–inverse Document Frequency *"In information retrieval, tf–idf, TF*IDF, or TFIDF, short for term frequency–inverse document frequency, is a numerical statistic that is intended to reflect how important a word is to a document in a collection or corpus.[1] It is often used as a weighting factor in searches of information retrieval, text mining, and user modeling. The tf–idf value increases proportionally to the number of times a word appears in the document and is offset by the number of documents in the corpus that contain the word, which helps to adjust for the fact that some words appear more frequently in general."*

Source: https://en.wikipedia.org/wiki/Tf%E2%80%93idf

tensors The main data structure of PyTorch. the tensor is an array. A vector is a 1-dimensional tensor, a matrix is a 2-dimensional tensor, an array with three indices is a 3-dimensional tensor (RGB color images for example)

torchvision The torchvision package consists of popular datasets, model architectures, and common image transformations for computer vision.

See torchvision[41]

Underwriter *"Insurance underwriters evaluate the risk and exposures of potential clients. They decide how much coverage the client should receive, how much they should pay for it, or whether even to accept the risk and insure them. Underwriting involves a measuring risk exposure and determining the premium that needs to be charged to insure that risk."*

See https://en.wikipedia.org/wiki/Underwriting#Insurance_underwriting

Source: Wikipedia contributors. (2019, August 9). Underwriting. In Wikipedia, The Free Encyclopedia. Retrieved 08:26, September 13, 2019, from https://en.wikipedia.org/w/index.php?title=Underwriting&oldid=910020948

Validation the action of checking or proving the validity or accuracy of the model generated by the Artificial Intelligence

[41] https://pytorch.org/docs/stable/torchvision/index.html

Validation Dataset The sample of data used to provide an unbiased evaluation of a model fit on the training dataset while tuning model hyperparameters. The evaluation becomes more biased as skill on the validation dataset is incorporated into the model configuration.

See About Train, Validation and Test Sets in Machine Learning[42]

Web Scraping Web scraping, web harvesting, or web data extraction is data scraping used for extracting data from websites. Web scraping software may access the World Wide Web directly using the Hypertext Transfer Protocol, or through a web browser.

Source: https://en.wikipedia.org/wiki/Web_scraping

13 Secure and Private AI : e-learning

e-learning from large platforms : https://udacity.com

13.1 Courses

- private ai : https://classroom.udacity.com/courses/ud185

13.2 release as a target type

```
git checkout master; git pull origin master
standard-version --first-release
standard-version
```

13.2.1 git add push

```
npm run release -- --release-as patch | minor | major
git push --follow-tags origin master
```

13.3 Run Jupyter notebooks

- On windows : `start > Ananconda prompt (miniconda2)`
- activate the environment `conda activiate py37`
- deactivate the environment `conda deactivate`
- determining my environment : `conda info --envs`

14 Secure and Private AI : Part 1 - Tensors in PyTorch-Exercises

Source : https://classroom.udacity.com/courses/ud185/lessons/8a993162-65c4-4a80-bd35-47d9f3a6f5bc/concepts/70526adf-40d3-4446-ac32-d3f798739745

[1]:
```
import torch
```

[42] https://towardsdatascience.com/train-validation-and-test-sets-72cb40cba9e7

```
[2]: def activation(x):
         """ Sigmoid activation function

             Arguments
             ---------
             x: torch.Tensor
         """
         return 1/(1+torch.exp(-x))
```

```
[3]: ### Generate some data
     torch.manual_seed(7) # Set the random seed so things are predictable

     # Features are 3 random normal variables
     features = torch.randn((1, 5))
     # True weights for our data, random normal variables again
     weights = torch.randn_like(features)
     # and a true bias term
     bias = torch.randn((1, 1))
```

```
[4]: print(features)
```

```
tensor([[-0.1468,  0.7861,  0.9468, -1.1143,  1.6908]])
```

```
[5]: print(weights)
```

```
tensor([[-0.8948, -0.3556,  1.2324,  0.1382, -1.6822]])
```

```
[6]: print(bias)
```

```
tensor([[0.3177]])
```

14.1 Exercise:

Calculate the output of the network with input features features, weights weights, and bias bias. Similar to Numpy, PyTorch has a torch.sum() function, as well as a .sum() method on tensors, for taking sums. Use the function activation defined above as the activation function.

```
[7]: y = activation(torch.sum(features * weights) + bias)
```

```
[8]: print (y)
```

```
tensor([[0.1595]])
```

```
[9]: #does not work because the size of features and weights aren't allowing the␣
     ↪multiplication

     # uncomment to test it torch.mm(features, weights)
```

```
[10]: features.shape
```

```
[10]: torch.Size([1, 5])
```

```
[11]: weights.shape
```

```
[11]: torch.Size([1, 5])
```

```
[12]: bias.shape
```

[12]:
```
torch.Size([1, 1])
```

14.2 Exercise:

Calculate the output of our little network using matrix multiplication. Reshape weights to have five rows and one column with something like weights.view(5, 1).

[13]:
```
activation(torch.mm(features, weights.view(5,1))+bias)
```

[13]:
```
tensor([[0.1595]])
```

[14]:
```
features.shape
```

[14]:
```
torch.Size([1, 5])
```

[15]:
```
### Generate some data
torch.manual_seed(7) # Set the random seed so things are predictable

# Features are 3 random normal variables
features = torch.randn((1, 3))

# Define the size of each layer in our network
n_input = features.shape[1]     # Number of input units, must match number of␣
↪input features
n_hidden = 2                    # Number of hidden units
n_output = 1                    # Number of output units

# Weights for inputs to hidden layer
W1 = torch.randn(n_input, n_hidden)
# Weights for hidden layer to output layer
W2 = torch.randn(n_hidden, n_output)

# and bias terms for hidden and output layers
B1 = torch.randn((1, n_hidden))
B2 = torch.randn((1, n_output))

print(W1)

print(W2)

print(B1)

print(B2)
```

```
tensor([[-1.1143,  1.6908],
        [-0.8948, -0.3556],
        [ 1.2324,  0.1382]])
tensor([[-1.6822],
        [ 0.3177]])
tensor([[0.1328, 0.1373]])
tensor([[0.2405]])
```

14.3 Exercise:

Calculate the output for this multi-layer network using the weights W1 & W2, and the biases, B1 & B2.

```
[16]: h = activation(torch.mm(features, W1) + B1)
      output = activation(torch.mm(h, W2) + B2)
      print(output)
```

```
tensor([[0.3171]])
```

```
[17]: import numpy as np
      a = np.random.rand(4,3)
      a
```

```
[17]: array([[0.98843161, 0.91529327, 0.87711017],
             [0.03036671, 0.87467354, 0.38863948],
             [0.15188762, 0.30653778, 0.87092012],
             [0.65136106, 0.59484865, 0.13040916]])
```

```
[18]: # To create a tensor from a Numpy array, use torch.from_numpy()

      b = torch.from_numpy(a)
      b
```

```
[18]: tensor([[0.9884, 0.9153, 0.8771],
              [0.0304, 0.8747, 0.3886],
              [0.1519, 0.3065, 0.8709],
              [0.6514, 0.5948, 0.1304]], dtype=torch.float64)
```

```
[19]: # To convert a tensor to a Numpy array, use the .numpy() method
      b.numpy()
```

```
[19]: array([[0.98843161, 0.91529327, 0.87711017],
             [0.03036671, 0.87467354, 0.38863948],
             [0.15188762, 0.30653778, 0.87092012],
             [0.65136106, 0.59484865, 0.13040916]])
```

```
[20]: # Multiply PyTorch Tensor by 2, in place
      b.mul_(2)
```

```
[20]: tensor([[1.9769, 1.8306, 1.7542],
              [0.0607, 1.7493, 0.7773],
              [0.3038, 0.6131, 1.7418],
              [1.3027, 1.1897, 0.2608]], dtype=torch.float64)
```

```
[21]: # Numpy array matches new values from Tensor
      a
```

```
[21]: array([[1.97686322, 1.83058654, 1.75422035],
             [0.06073342, 1.74934708, 0.77727897],
             [0.30377524, 0.61307556, 1.74184025],
             [1.30272212, 1.18969731, 0.26081832]])
```

15 Secure and Private AI : Part 2 - Neural Networks in PyTorch (Exercises)

Source : https://classroom.udacity.com/courses/ud185/lessons/8a993162-65c4-4a80-bd35-47d9f3a6f5bc/concepts/70526adf-40d3-4446-ac32-d3f798739745#

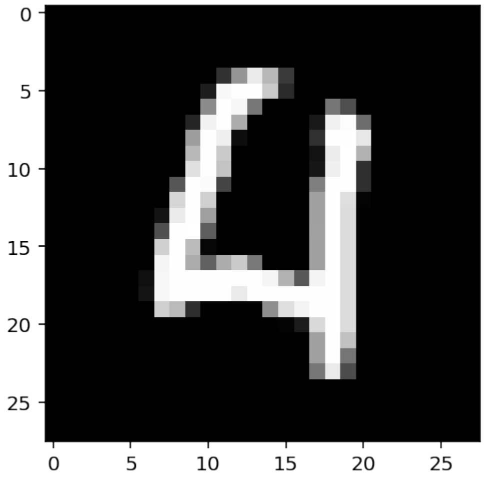
[69]:
```
# Import necessary packages

%matplotlib inline
%config InlineBackend.figure_format = 'retina'

import numpy as np
import torch

import helper

import matplotlib.pyplot as plt
```

[70]:
```
# download images including figures to recognize

from torchvision import datasets, transforms

# Define a transform to normalize the data
transform = transforms.Compose([transforms.ToTensor(),
                              transforms.Normalize((0.5,), (0.5,)),
                              ])

# Download and load the training data
trainset = datasets.MNIST('~/.pytorch/MNIST_data/', download=True, train=True,␣
↪transform=transform)
trainloader = torch.utils.data.DataLoader(trainset, batch_size=64, shuffle=True)
```

[71]:
```
dataiter = iter(trainloader)
images, labels = dataiter.next()
print('type(images) : ')
print(type(images))
print('images.shape : ')
print(images.shape)
print('labels.shape : ')
print(labels.shape)
```

```
type(images) :
<class 'torch.Tensor'>
images.shape :
torch.Size([64, 1, 28, 28])
labels.shape :
torch.Size([64])
```

[72]:
```
# display one image

plt.imshow(images[1].numpy().squeeze(), cmap='Greys_r');
```



15.1 Exercise:

1. Flatten the batch of images images. See https://pytorch.org/docs/stable/torch.html#torch.flatten
2. Then build a multi-layer network with 784 input units, 256 hidden units, and 10 output units using random tensors for the weights and biases.

For now, use a sigmoid activation for the hidden layer. Leave the output layer without an activation, we'll add one that gives us a probability distribution next.

[1]:

```
## Your solution

def activation(x):
    return 1/(1+torch.exp(-x))

# inputs = torch.flatten(images)
inputs = images.view(images.shape[0], -1)
# Create parameters
w1 = torch.randn(784, 256)
b1 = torch.randn(256)

w2 = torch.randn(256, 10)
b2 = torch.randn(10)

h = activation(torch.mm(inputs, w1) + b1)

out = torch.mm(h, w2) + b2 # output of your network, should have shape (64,10)
print(out)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-1-92f80f8bbe6b> in <module>
      5
      6 # inputs = torch.flatten(images)
----> 7 inputs = images.view(images.shape[0], -1)
      8 # Create parameters
      9 w1 = torch.randn(784, 256)

NameError: name 'images' is not defined
```

15.2 Exercise:

1. Implement a function softmax that performs the softmax calculation and returns probability distributions for each example in the batch.

Note that you'll need to pay attention to the shapes when doing this.

A. If you have a tensor a with shape (64, 10) and a tensor b with shape (64,), doing a/b will give you an error because PyTorch will try to do the division across the columns (called broadcasting) but you'll get a size mismatch.

The way to think about this is for each of the 64 examples, you only want to divide by one value, the sum in the denominator.

So you need b to have a shape of (64, 1).

This way PyTorch will divide the 10 values in each row of 'a' by the one value in each row of 'b'. ' Pay attention to how you take the sum as well.

1. You'll need to define the dim keyword in torch.sum.
2. Setting dim=0 takes the sum across the rows while dim=1 takes the sum across the columns.

[74]:

```
def softmax(x):
    return torch.exp(x)/torch.sum(torch.exp(x), dim=1).view(-1, 1)

# Here, out should be the output of the network in the previous excercise with␣
↪shape (64,10)
probabilities = softmax(out)

# Does it have the right shape? Should be (64, 10)
print(probabilities.shape)
# Does it sum to 1?
print(probabilities.sum(dim=1))
```

```
torch.Size([64, 10])
tensor([1.0000, 1.0000, 1.0000, 1.0000, 1.0000, 1.0000, 1.0000, 1.0000, 1.0000,
        1.0000, 1.0000, 1.0000, 1.0000, 1.0000, 1.0000, 1.0000, 1.0000, 1.0000,
        1.0000, 1.0000, 1.0000, 1.0000, 1.0000, 1.0000, 1.0000, 1.0000, 1.0000,
        1.0000, 1.0000, 1.0000, 1.0000, 1.0000, 1.0000, 1.0000, 1.0000, 1.0000,
        1.0000, 1.0000, 1.0000, 1.0000, 1.0000, 1.0000, 1.0000, 1.0000, 1.0000,
        1.0000, 1.0000, 1.0000, 1.0000, 1.0000, 1.0000, 1.0000, 1.0000, 1.0000,
        1.0000, 1.0000, 1.0000, 1.0000, 1.0000, 1.0000, 1.0000, 1.0000, 1.0000,
        1.0000])
```

15.3 Building networks with PyTorch

PyTorch provides a module nn that makes building networks much simpler.

Here I'll show you how to build the same one as above with 784 inputs, 256 hidden units, 10 output units and a softmax output.

[75]:

```
from torch import nn
```

[76]:

```
class Network(nn.Module):
    def __init__(self):
        super().__init__()

        # Inputs to hidden layer linear transformation
        self.hidden = nn.Linear(784, 256)
        # Output layer, 10 units - one for each digit
        self.output = nn.Linear(256, 10)

```

(continues on next page)

(continued from previous page)

```
        # Define sigmoid activation and softmax output
        self.sigmoid = nn.Sigmoid()
        self.softmax = nn.Softmax(dim=1)

    def forward(self, x):
        # Pass the input tensor through each of our operations
        x = self.hidden(x)
        x = self.sigmoid(x)
        x = self.output(x)
        x = self.softmax(x)

        return x
```

[77]:

```
# Create the network and look at it's text representation
model = Network()
model
```

[77]:

```
Network(
  (hidden): Linear(in_features=784, out_features=256, bias=True)
  (output): Linear(in_features=256, out_features=10, bias=True)
  (sigmoid): Sigmoid()
  (softmax): Softmax(dim=1)
)
```

[78]:

```
import torch.nn.functional as F

class Network(nn.Module):
    def __init__(self):
        super().__init__()
        # Inputs to hidden layer linear transformation
        self.hidden = nn.Linear(784, 256)
        # Output layer, 10 units - one for each digit
        self.output = nn.Linear(256, 10)

    def forward(self, x):
        # Hidden layer with sigmoid activation
        x = F.sigmoid(self.hidden(x))
        # Output layer with softmax activation
        x = F.softmax(self.output(x), dim=1)

        return x
```

15.4 Exercise:

1. Create a network with 784 input units, a hidden layer with 128 units and a ReLU activation,
2. then a hidden layer with 64 units and a ReLU activation,
3. and finally an output layer with a softmax activation as shown above.

You can use a ReLU activation with the nn.ReLU module or F.relu function.

HINT : It's good practice to name your layers by their type of network, for instance 'fc' to represent a fully-connected layer. As you code your solution, use fc1, fc2, and fc3 as your layer names.

[79]:

```
class Network(nn.Module):
    def __init__(self):
        super().__init__()
        # Inputs to hidden layer linear transformation
        self.fc1 = nn.Linear(784, 128)
```

(continues on next page)

(continued from previous page)

```
        # Output layer, 10 units - one for each digit
        self.fc2 = nn.Linear(128, 64)
        self.fc3 = nn.Linear(64, 10)

    def forward(self, x):
        # Hidden layer with sigmoid activation
        x = self.fc1(x)
        x = F.relu(x)
        x = self.fc2(x)
        x = F.relu(x)
        x = self.fc3(x)
        x = F.softmax(x, dim=1)

        return x

model = Network()
model
```

[79]:

```
Network(
  (fc1): Linear(in_features=784, out_features=128, bias=True)
  (fc2): Linear(in_features=128, out_features=64, bias=True)
  (fc3): Linear(in_features=64, out_features=10, bias=True)
)
```

[80]:

```
print(model.fc1.weight)
print(model.fc1.bias)
```

```
Parameter containing:
tensor([[ 0.0275,  0.0114, -0.0200,  ...,  0.0082,  0.0105,  0.0013],
        [-0.0329, -0.0207,  0.0299,  ..., -0.0202,  0.0338,  0.0142],
        [ 0.0187,  0.0115,  0.0148,  ...,  0.0082, -0.0069, -0.0235],
        ...,
        [-0.0160, -0.0043,  0.0133,  ...,  0.0289, -0.0234, -0.0295],
        [ 0.0008,  0.0087, -0.0107,  ...,  0.0169,  0.0172,  0.0253],
        [-0.0172, -0.0169,  0.0292,  ..., -0.0216, -0.0117,  0.0242]],
       requires_grad=True)
Parameter containing:
tensor([-0.0087, -0.0065, -0.0074,  0.0183,  0.0198,  0.0112, -0.0180,  0.0023,
        -0.0313, -0.0017,  0.0059,  0.0179, -0.0348, -0.0271,  0.0289, -0.0218,
        -0.0349,  0.0230,  0.0251,  0.0121,  0.0112, -0.0171, -0.0222, -0.0187,
         0.0180, -0.0164,  0.0103,  0.0128,  0.0199,  0.0274,  0.0249, -0.0057,
        -0.0093,  0.0261,  0.0092,  0.0072, -0.0179,  0.0306, -0.0021, -0.0200,
         0.0217,  0.0253,  0.0149, -0.0018, -0.0171, -0.0097,  0.0106,  0.0302,
         0.0026,  0.0095, -0.0190,  0.0097,  0.0334, -0.0355, -0.0118, -0.0265,
        -0.0155, -0.0196,  0.0328,  0.0173,  0.0188,  0.0317,  0.0127,  0.0187,
        -0.0241,  0.0153, -0.0003,  0.0352, -0.0152, -0.0103, -0.0305, -0.0332,
         0.0262,  0.0242, -0.0225,  0.0266,  0.0343, -0.0306,  0.0123, -0.0033,
        -0.0104,  0.0064,  0.0349,  0.0114,  0.0083,  0.0334,  0.0024, -0.0074,
         0.0329,  0.0272, -0.0017, -0.0021, -0.0235, -0.0232, -0.0312, -0.0278,
        -0.0285, -0.0110, -0.0268,  0.0039, -0.0168, -0.0108, -0.0108, -0.0062,
        -0.0126,  0.0058, -0.0229, -0.0105, -0.0253, -0.0287,  0.0221,  0.0140,
        -0.0069,  0.0259, -0.0159,  0.0044, -0.0226,  0.0067,  0.0117, -0.0033,
        -0.0150,  0.0202,  0.0208, -0.0199, -0.0324,  0.0187, -0.0048, -0.0269],
       requires_grad=True)
```

[81]:

```
# Set biases to all zeros
model.fc1.bias.data.fill_(0)
```

[81]:

```
tensor([0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
↪ 0., 0., 0., 0., 0.,
        0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
↪ 0., 0., 0., 0., 0.,
```

(continues on next page)

(continued from previous page)

```
        0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
↪ 0., 0., 0., 0., 0.,
        0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
↪ 0., 0., 0., 0., 0.,
        0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
↪ 0., 0., 0., 0., 0.,
        0., 0., 0., 0., 0., 0., 0., 0.])
```

[82]:

```
print(model.fc1.weight)
print(model.fc1.bias)
```

```
Parameter containing:
tensor([[ 0.0275,  0.0114, -0.0200,  ...,  0.0082,  0.0105,  0.0013],
        [-0.0329, -0.0207,  0.0299,  ..., -0.0202,  0.0338,  0.0142],
        [ 0.0187,  0.0115,  0.0148,  ...,  0.0082, -0.0069, -0.0235],
        ...,
        [-0.0160, -0.0043,  0.0133,  ...,  0.0289, -0.0234, -0.0295],
        [ 0.0008,  0.0087, -0.0107,  ...,  0.0169,  0.0172,  0.0253],
        [-0.0172, -0.0169,  0.0292,  ..., -0.0216, -0.0117,  0.0242]],
       requires_grad=True)
Parameter containing:
tensor([0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
↪ 0., 0., 0., 0., 0.,
        0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
↪ 0., 0., 0., 0., 0.,
        0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
↪ 0., 0., 0., 0., 0.,
        0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
↪ 0., 0., 0., 0., 0.,
        0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
↪ 0., 0., 0., 0., 0.,
        0., 0., 0., 0., 0., 0., 0., 0.], requires_grad=True)
```

[83]:

```
# sample from random normal with standard dev = 0.01
model.fc1.weight.data.normal_(std=0.01)
```

[83]:

```
tensor([[-0.0078,  0.0095, -0.0036,  ...,  0.0071,  0.0146,  0.0063],
        [ 0.0010,  0.0006, -0.0001,  ..., -0.0100,  0.0074,  0.0048],
        [-0.0031,  0.0043,  0.0034,  ...,  0.0015,  0.0080, -0.0077],
        ...,
        [-0.0044,  0.0021, -0.0046,  ...,  0.0097, -0.0094,  0.0094],
        [ 0.0127, -0.0035, -0.0029,  ..., -0.0130,  0.0051,  0.0059],
        [ 0.0003,  0.0035,  0.0014,  ...,  0.0022,  0.0069,  0.0070]])
```

15.5 Forward pass

Now that we have a network, let's see what happens when we pass in an image.

As you can see above, our network has basically no idea what this digit is.

It's because we haven't trained it yet, all the weights are random!

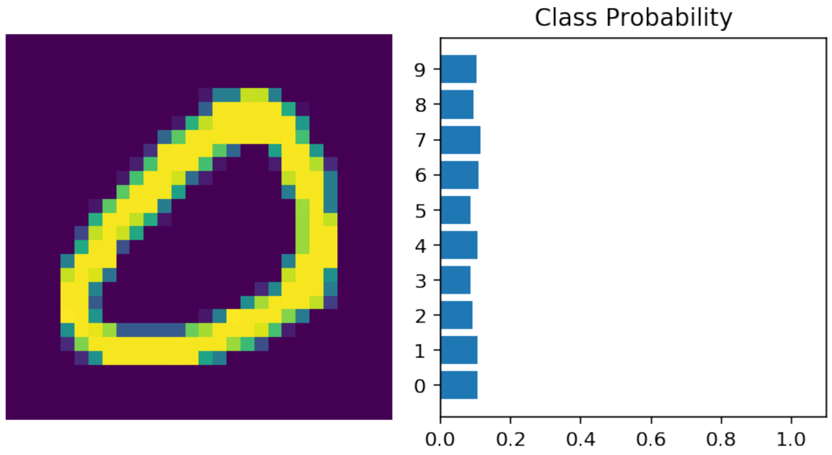
[84]:

```
# Grab some data
dataiter = iter(trainloader)
images, labels = dataiter.next()

# Resize images into a 1D vector, new shape is (batch size, color channels, image␣
↪pixels)
images.resize_(64, 1, 784)
# or images.resize_(images.shape[0], 1, 784) to automatically get batch size
```

(continues on next page)

(continued from previous page)

```
# Forward pass through the network
img_idx = 0
ps = model.forward(images[img_idx,:])

img = images[img_idx]
helper.view_classify(img.view(1, 28, 28), ps)
```



As you can see above, our network has basically no idea what this digit is. It's because we haven't trained it yet, all the weights are random!

15.5.1 Using `nn.Sequential`

PyTorch provides a convenient way to build networks like this where a tensor is passed sequentially through operations, `nn.Sequential` (documentation[43]). Using this to build the equivalent network:

```
[85]: # Hyperparameters for our network
input_size = 784
hidden_sizes = [128, 64]
output_size = 10

# Build a feed-forward network
model = nn.Sequential(nn.Linear(input_size, hidden_sizes[0]),
                      nn.ReLU(),
                      nn.Linear(hidden_sizes[0], hidden_sizes[1]),
                      nn.ReLU(),
                      nn.Linear(hidden_sizes[1], output_size),
                      nn.Softmax(dim=1))
print(model)

# Forward pass through the network and display output
images, labels = next(iter(trainloader))
images.resize_(images.shape[0], 1, 784)
ps = model.forward(images[0,:])
helper.view_classify(images[0].view(1, 28, 28), ps)
```

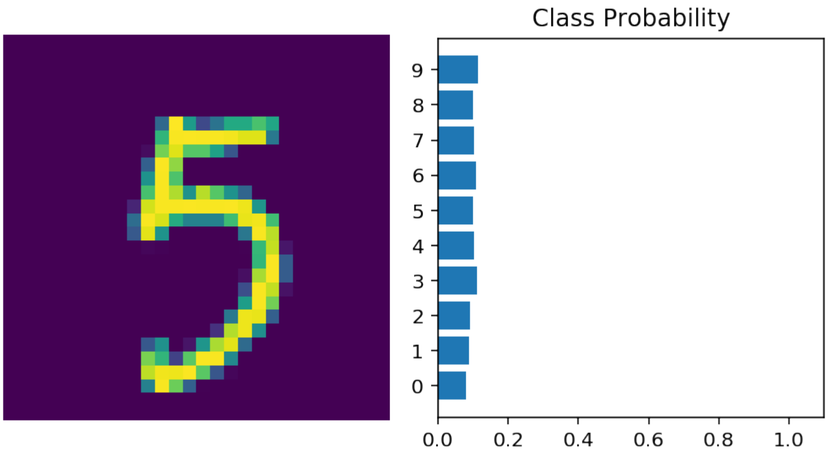
```
Sequential(
  (0): Linear(in_features=784, out_features=128, bias=True)
  (1): ReLU()
  (2): Linear(in_features=128, out_features=64, bias=True)
  (3): ReLU()
  (4): Linear(in_features=64, out_features=10, bias=True)
```

(continues on next page)

[43] https://pytorch.org/docs/master/nn.html#torch.nn.Sequential

(continued from previous page)

```
  (5): Softmax(dim=1)
)
```



```
[86]: print('model[0] :')
      print(model[0])
      print('model[0].weight : ')
      model[0].weight
```

```
model[0] :
Linear(in_features=784, out_features=128, bias=True)
model[0].weight :
```

```
[86]: Parameter containing:
      tensor([[-0.0265,  0.0257,  0.0099,  ..., -0.0191, -0.0272, -0.0022],
              [ 0.0045, -0.0074, -0.0298,  ..., -0.0334, -0.0355,  0.0267],
              [ 0.0127, -0.0166, -0.0050,  ..., -0.0306, -0.0041, -0.0212],
              ...,
              [ 0.0288, -0.0229,  0.0030,  ..., -0.0190, -0.0303,  0.0076],
              [-0.0103, -0.0052,  0.0308,  ...,  0.0225, -0.0185,  0.0263],
              [-0.0266,  0.0273,  0.0299,  ...,  0.0159,  0.0048,  0.0118]],
             requires_grad=True)
```

You can also pass in an OrderedDict to name the individual layers and operations, instead of using incremental integers. Note that dictionary keys must be unique, so each operation must have a different name.

```
[87]: from collections import OrderedDict
      model = nn.Sequential(OrderedDict([
                            ('fc1', nn.Linear(input_size, hidden_sizes[0])),
                            ('relu1', nn.ReLU()),
                            ('fc2', nn.Linear(hidden_sizes[0], hidden_sizes[1])),
                            ('relu2', nn.ReLU()),
                            ('output', nn.Linear(hidden_sizes[1], output_size)),
                            ('softmax', nn.Softmax(dim=1))]))
      model
```

```
[87]: Sequential(
        (fc1): Linear(in_features=784, out_features=128, bias=True)
        (relu1): ReLU()
        (fc2): Linear(in_features=128, out_features=64, bias=True)
        (relu2): ReLU()
        (output): Linear(in_features=64, out_features=10, bias=True)
        (softmax): Softmax(dim=1)
      )
```

[88]:

```
print(model[0])
print(model.fc1)
```

```
Linear(in_features=784, out_features=128, bias=True)
Linear(in_features=784, out_features=128, bias=True)
```

In the next notebook, we'll see how we can train a neural network to accuractly predict the numbers appearing in the MNIST images.

16 Secure and Private AI : Part 3 - Training Neural Networks (Solutions)

[1]:

```
import torch
from torch import nn
import torch.nn.functional as F
from torchvision import datasets, transforms

# Define a transform to normalize the data
transform = transforms.Compose([transforms.ToTensor(),
                                transforms.Normalize((0.5, ), (0.5, )),
                              ])
# Download and load the training data
trainset = datasets.MNIST('~/.pytorch/MNIST_data/', download=True, train=True,␣
↪transform=transform)
trainloader = torch.utils.data.DataLoader(trainset, batch_size=64, shuffle=True)
```

[2]:

```
# Build a feed-forward network
model = nn.Sequential(nn.Linear(784, 128),
                      nn.ReLU(),
                      nn.Linear(128, 64),
                      nn.ReLU(),
                      nn.Linear(64, 10))

# Define the loss
criterion = nn.CrossEntropyLoss()
print('criterion')
print(criterion)

# Get our data
images, labels = next(iter(trainloader))
# Flatten images
images = images.view(images.shape[0], -1)

print(images)
# Forward pass, get our logits
logits = model(images)
# Calculate the loss with the logits and the labels
loss = criterion(logits, labels)

print(loss)
```

```
criterion
CrossEntropyLoss()
tensor([[-1., -1., -1.,  ..., -1., -1., -1.],
        [-1., -1., -1.,  ..., -1., -1., -1.],
        [-1., -1., -1.,  ..., -1., -1., -1.],
        ...,
        [-1., -1., -1.,  ..., -1., -1., -1.],
```

(continues on next page)

(continued from previous page)

```
        [-1., -1., -1.,  ..., -1., -1., -1.],
        [-1., -1., -1.,  ..., -1., -1., -1.]])
tensor(2.3126, grad_fn=<NllLossBackward>)
```

16.1 Exercise:

Build a model that returns the log-softmax as the output and calculate the loss using the negative log likelihood loss.

[3]:

```
# Build a feed-forward network
model = nn.Sequential(nn.Linear(784, 128),
                      nn.ReLU(),
                      nn.Linear(128, 64),
                      nn.ReLU(),
                      nn.Linear(64, 10),
                      nn.LogSoftmax(dim=1)
                     )

# Define the loss
criterion = nn.NLLLoss()
print('criterion')
print(criterion)

# Get our data
images, labels = next(iter(trainloader))
# Flatten images
images = images.view(images.shape[0], -1)

print(images)
# Forward pass, get our logits
logits = model(images)
# Calculate the loss with the logits and the labels
loss = criterion(logits, labels)

print(loss)
```

```
criterion
NLLLoss()
tensor([[-1., -1., -1.,  ..., -1., -1., -1.],
        [-1., -1., -1.,  ..., -1., -1., -1.],
        [-1., -1., -1.,  ..., -1., -1., -1.],
        ...,
        [-1., -1., -1.,  ..., -1., -1., -1.],
        [-1., -1., -1.,  ..., -1., -1., -1.],
        [-1., -1., -1.,  ..., -1., -1., -1.]])
tensor(2.3272, grad_fn=<NllLossBackward>)
```

16.2 Autograd

Torch provides a module, autograd, for automatically calculating the gradients of tensors. We can use it to calculate the gradients of all our parameters with respect to the loss.

[4]:

```
x = torch.randn(2,2, requires_grad=True)
print(x)
```

```
tensor([[-0.3005,  1.1679],
        [ 0.0308,  0.6812]], requires_grad=True)
```

```
[5]: y = x**2
     print(y)
```

```
tensor([[9.0282e-02, 1.3641e+00],
        [9.4915e-04, 4.6403e-01]], grad_fn=<PowBackward0>)
```

```
[6]: ## grad_fn shows the function that generated this variable
     print(y.grad_fn)
```

```
<PowBackward0 object at 0x000002550F873B08>
```

```
[7]: z = y.mean()
     print(z)
```

```
tensor(0.4798, grad_fn=<MeanBackward0>)
```

```
[8]: print(x.grad)
```

```
None
```

```
[9]: z.backward()
     print(x.grad)
     print(x/2)
```

```
tensor([[-0.1502,  0.5840],
        [ 0.0154,  0.3406]])
tensor([[-0.1502,  0.5840],
        [ 0.0154,  0.3406]], grad_fn=<DivBackward0>)
```

16.3 Loss and Autograd together

When we create a network with PyTorch, all of the parameters are initialized with requires_grad = True.

This means that when we calculate the loss and call loss.backward(), the gradients for the parameters are calculated.

These gradients are used to update the weights with gradient descent.

Below you can see an example of calculating the gradients using a backwards pas

```
[10]: # Build a feed-forward network
      model = nn.Sequential(nn.Linear(784, 128),
                            nn.ReLU(),
                            nn.Linear(128, 64),
                            nn.ReLU(),
                            nn.Linear(64, 10),
                            nn.LogSoftmax(dim=1))

      criterion = nn.NLLLoss()
      images, labels = next(iter(trainloader))
      images = images.view(images.shape[0], -1)

      logps = model(images)
      loss = criterion(logps, labels)
```

```
[11]: print('Before backward pass: \n', model[0].weight.grad)

      loss.backward()

      print('After backward pass: \n', model[0].weight.grad)
```

```
Before backward pass:
 None
After backward pass:
 tensor([[-0.0004, -0.0004, -0.0004,  ..., -0.0004, -0.0004, -0.0004],
        [-0.0013, -0.0013, -0.0013,  ..., -0.0013, -0.0013, -0.0013],
        [ 0.0000,  0.0000,  0.0000,  ...,  0.0000,  0.0000,  0.0000],
        ...,
        [-0.0007, -0.0007, -0.0007,  ..., -0.0007, -0.0007, -0.0007],
        [ 0.0016,  0.0016,  0.0016,  ...,  0.0016,  0.0016,  0.0016],
        [-0.0010, -0.0010, -0.0010,  ..., -0.0010, -0.0010, -0.0010]])
```

16.4 Training the network

```
[12]: from torch import optim

      # Optimizers require the parameters to optimize and a learning rate
      optimizer = optim.SGD(model.parameters(), lr=0.01)
```

```
[13]: print('Initial weights - ', model[0].weight)

      images, labels = next(iter(trainloader))
      images.resize_(64, 784)

      # Clear the gradients, do this because gradients are accumulated
      optimizer.zero_grad()

      # Forward pass, then backward pass, then update weights
      output = model.forward(images)
      loss = criterion(output, labels)
      loss.backward()
      print('Gradient -', model[0].weight.grad)
```

```
Initial weights -  Parameter containing:
tensor([[ 0.0032, -0.0306,  0.0024,  ...,  0.0189,  0.0254, -0.0141],
        [-0.0143,  0.0241, -0.0322,  ..., -0.0284, -0.0339,  0.0028],
        [ 0.0268,  0.0306, -0.0296,  ...,  0.0029, -0.0179,  0.0179],
        ...,
        [ 0.0035, -0.0009, -0.0033,  ...,  0.0285,  0.0051, -0.0294],
        [-0.0066,  0.0179, -0.0060,  ...,  0.0012, -0.0074,  0.0304],
        [-0.0297, -0.0225, -0.0018,  ...,  0.0074,  0.0031,  0.0257]],
       requires_grad=True)
Gradient - tensor([[ 0.0013,  0.0013,  0.0013,  ...,  0.0013,  0.0013,  0.0013],
        [ 0.0015,  0.0015,  0.0015,  ...,  0.0015,  0.0015,  0.0015],
        [ 0.0000,  0.0000,  0.0000,  ...,  0.0000,  0.0000,  0.0000],
        ...,
        [-0.0002, -0.0002, -0.0002,  ..., -0.0002, -0.0002, -0.0002],
        [ 0.0012,  0.0012,  0.0012,  ...,  0.0012,  0.0012,  0.0012],
        [-0.0002, -0.0002, -0.0002,  ..., -0.0002, -0.0002, -0.0002]])
```

```
[14]: # Take an update step and few the new weights
      optimizer.step()
      print('Updated weights - ', model[0].weight)
```

```
Updated weights -  Parameter containing:
tensor([[ 0.0031, -0.0306,  0.0023,  ...,  0.0189,  0.0253, -0.0141],
        [-0.0143,  0.0241, -0.0322,  ..., -0.0284, -0.0339,  0.0028],
        [ 0.0268,  0.0306, -0.0296,  ...,  0.0029, -0.0179,  0.0179],
        ...,
        [ 0.0035, -0.0009, -0.0033,  ...,  0.0285,  0.0051, -0.0294],
        [-0.0066,  0.0179, -0.0060,  ...,  0.0012, -0.0074,  0.0304],
```

(continues on next page)

(continued from previous page)

```
        [-0.0297, -0.0225, -0.0018,  ...,  0.0074,  0.0031,  0.0257]],
       requires_grad=True)
```

16.5 Training for real

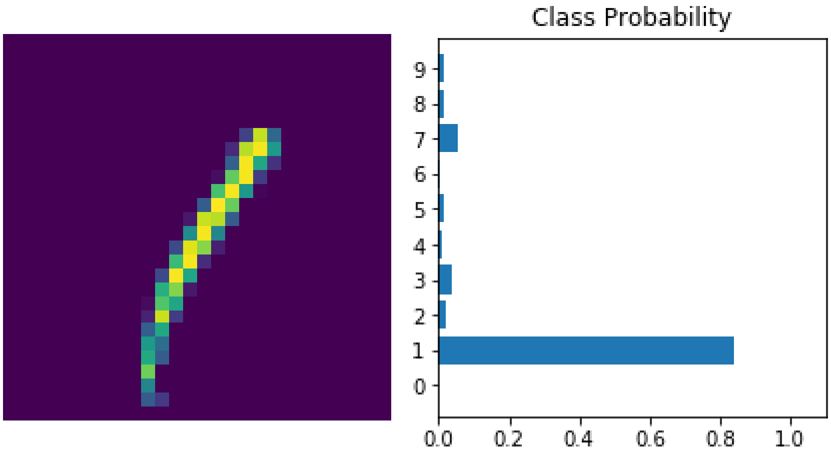
16.5.1 Exercise:

Implement the training pass for our network. If you implemented it correctly, you should see the training loss drop with each epoch.

```
[15]: model = nn.Sequential(nn.Linear(784, 128),
                            nn.ReLU(),
                            nn.Linear(128, 64),
                            nn.ReLU(),
                            nn.Linear(64, 10),
                            nn.LogSoftmax(dim=1))

      criterion = nn.NLLLoss()
      optimizer = optim.SGD(model.parameters(), lr=0.003)

      epochs = 5
      for e in range(epochs):
          running_loss = 0
          for images, labels in trainloader:
              # Flatten MNIST images into a 784 long vector
              images = images.view(images.shape[0], -1)

              # TODO: Training pass
              optimizer.zero_grad()

              output = model.forward(images)
              loss = criterion(output, labels)
              loss.backward()
              optimizer.step()

              running_loss += loss.item()
          else:
              print(f"Training loss: {running_loss/len(trainloader)}")
```

```
Training loss: 1.9068074593666011
Training loss: 0.8489092080387225
Training loss: 0.5150735600511911
Training loss: 0.42009466452829874
Training loss: 0.3772694453247575
```

```
[16]: %matplotlib inline
      import helper

      images, labels = next(iter(trainloader))

      img = images[0].view(1, 784)
      # Turn off gradients to speed up this part
      with torch.no_grad():
          logps = model.forward(img)

      # Output of the network are logits, need to take softmax for probabilities
      ps = torch.exp(logps)
      helper.view_classify(img.view(1, 28, 28), ps)
```



```
[ ]:
```

17 Secure and Private AI : Part 4 - Classifying Fashion-MNIST (Solution)

source : https://classroom.udacity.com/courses/ud185/lessons/8a993162-65c4-4a80-bd35-47d9f3a6f5bc/concepts/70526adf-40d3-4446-ac32-d3f798739745

```
[1]: import torch
from torchvision import datasets, transforms
import helper

# define a transformation to normalize the data
transform = transforms.Compose([transforms.ToTensor(), transforms.Normalize((0.5,␣
↪), (0.5,))])

# Download and load the training data
trainset = datasets.FashionMNIST('~/.pytorch/F_MNIST_data/', download=True,␣
↪transform=transform)
trainloader = torch.utils.data.DataLoader(trainset, batch_size=64, shuffle=True)

# Download and load the test data
testset = datasets.FashionMNIST('~/.pytorch/F_MNIST_data/', download=True,␣
↪train=False, transform=transform)
testloader = torch.utils.data.DataLoader(testset, batch_size=64, shuffle=True)
```

```
[2]: # see one of the images
image, label = next(iter(trainloader))
helper.imshow(image[0,:]);
```

[3]:

```
# building the network

from torch import nn, optim
import torch.nn.functional as F
```

[4]:

```
# define the network architecture
class Classifier(nn.Module):
    def __init__(self):
        super().__init__()
        self.fc1 = nn.Linear(784, 256)
        self.fc2 = nn.Linear(256, 128)
        self.fc3 = nn.Linear(128, 64)
        self.fc4 = nn.Linear(64, 10)

    def forward(self, x):
        #make sure the input tensor is flattened
        x = x.view(x.shape[0], -1)
        x = F.relu(self.fc1(x))
        x = F.relu(self.fc2(x))
        x = F.relu(self.fc3(x))
        x = F.log_softmax(self.fc4(x), dim=1)

        return x
```

17.1 Train the network

1. First you'll want to define the criterion (something like nn. or nn.NLLLoss)
2. and the optimizer (typically optim.SGD or optim.Adam).

Then write the training code. Remember the training pass is a fairly straightforward process:

- Make a forward pass through the network to get the logits
- Use the logits to calculate the loss
- Perform a backward pass through the network with `loss.backward()` to calculate the gradients
- Take a step with the optimizer to update the weights

By adjusting the hyperparameters (hidden units, learning rate, etc), you should be able to get the training loss below 0.4

[6]:

```
# create the network, define the criterion and optimizer

```

(continues on next page)

(continued from previous page)

```
model = Classifier()
criterion = nn.NLLLoss()
optimizer = optim.Adam(model.parameters(), lr=0.003)
```

17.2 Try to predict the clothes without training

[ ]:

```
%matplotlib inline
%config InlineBackend.figure_format = 'retina'

import helper

# test out our network!

dataiter = iter(testloader)
images, labels = dataiter.next()
img = images[1]

# calculate the class probabilities (softmax) for img
ps = torch.exp(model(img))

# plot the image and probabilities
helper.view_classify(img, ps, version='Fashion')
```

17.3 Now, train the model and increase the quality of the predictions!

[8]:

```
# train the model
epochs = 5
for e in range(epochs):
    running_loss = 0
    for images, labels in trainloader:
        log_ps = model(images)
        loss = criterion(log_ps, labels)

        optimizer.zero_grad()
        loss.backward()
        optimizer.step()

        running_loss += loss.item()

    else:
        print(f"training loss: {running_loss}")
```

```
training loss: 489.07935957610607
training loss: 365.2417050972581
training loss: 335.35467436909676
training loss: 312.6761027649045
training loss: 295.91014794260263
```

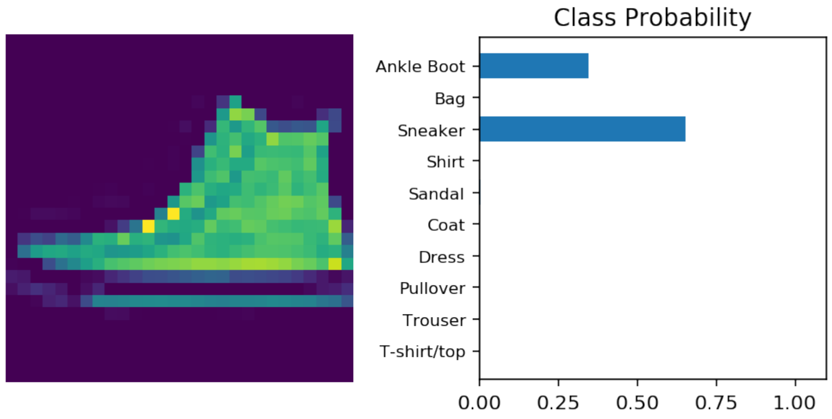
[9]:

```
%matplotlib inline
%config InlineBackend.figure_format = 'retina'

import helper

# test out our network!

dataiter = iter(testloader)
images, labels = dataiter.next()
```

(continues on next page)

(continued from previous page)

```
img = images[1]

# calculate the class probabilities (softmax) for img
ps = torch.exp(model(img))

# plot the image and probabilities
helper.view_classify(img, ps, version='Fashion')
```



[ ]:

18 Indices and tables

- genindex
- search

Index