



“nn” things
every Java
Developer
should know
about
AI/ML/DL

Mani Sarkar
@theNeomatrix369

Use Hashtag:
#jOnConf

About me

- Member of the *London Java Community*
- Java / JVM, Polyglot Developer
- Freelance Software, Data, ML Engineer
- Code quality, testing, performance, DevOps, deep affinity for AI/ML/DL, NN, topics
- *Strengthening* teams and helping them *accelerate*
- Involved in F/OSS i.e. GraalVM, AdoptOpenJDK and other projects and developer communities



Mani Sarkar
@theNeomatrix369

Use Hashtag:
#jOnConf

Also...

- Java Champion
- And Oracle Groundbreaker Ambassador

Click here [for more info on Freebies!](#)

Disclaimer

- Sharing my ideas, but *YMMV*
- Possibly missed one or more things or made mistakes, *I ask for forgiveness*
- Not clear for some, *my apologies*
- Lots of info and resources
- Lots of takeaway, please go home and do some more research
- Please contribute and share, please tweet!

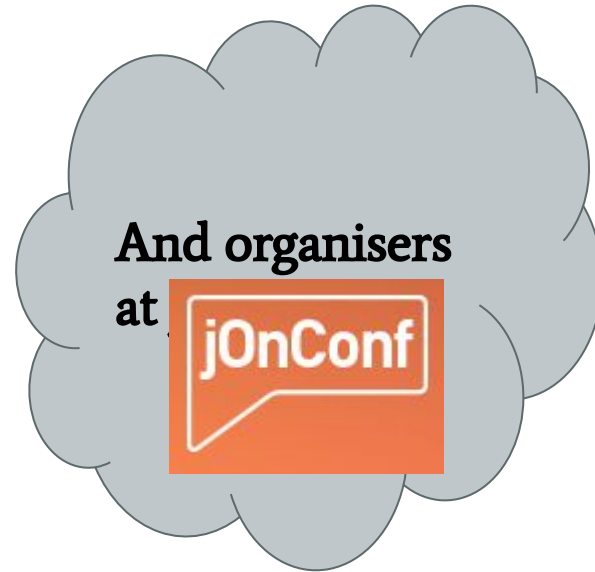
Get the slide deck (just now) at



<https://bit.ly/nn-things-java-dev-ai-ml-dl>

Thank you!

Martin Toshev and
Martin Patsov!



Use Hashtag:
#jOnConf

Agenda

Why “nn” things?

Demo, maybe
some code!

Timeline: how it
started for me...

Insights, shares
and tips...

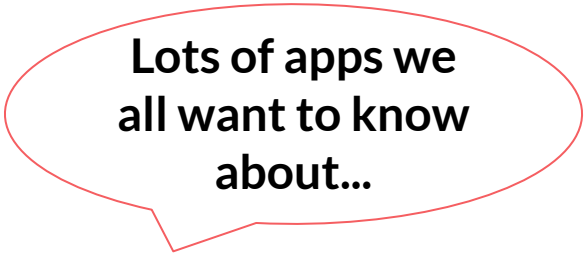
What we won't cover...

Pros or cons of ...

What is the best
library / framework to
do “xyz” in?

We may name some topics but we won't go into the details of them. We name them or express an opinion so we have a starting point in our learning journey!

Celebrating 25 years of Java



Lots of apps we
all want to know
about...

“The 25 greatest Java
apps ever written” blog
post by Alexa Weber
Morales

...[for more go here](#)

Why this talk or such
talks?

They are all doing
an awesome job
and a service to
us!

Many can show
how to install or
configure “xyz”!

Many have shown
us how to code
super fast and at a
presentation!

Many have talked
about frameworks,
APIs, algorithms,
theories, and the
works!

But not many talks
about how to get
there, how each one
of us can become
better?

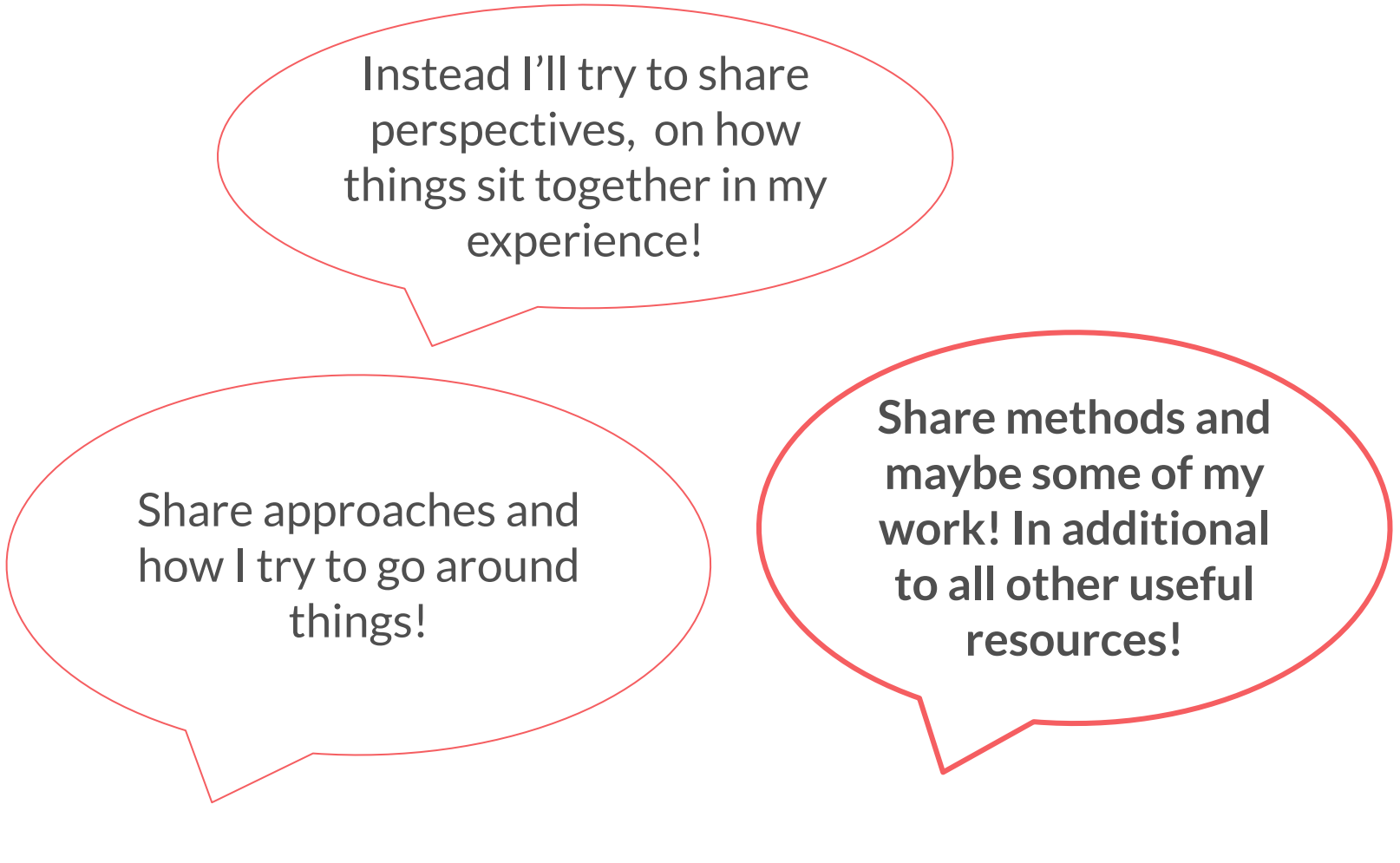


I could just show
you some code

Share some links
and videos

Show some
demos!

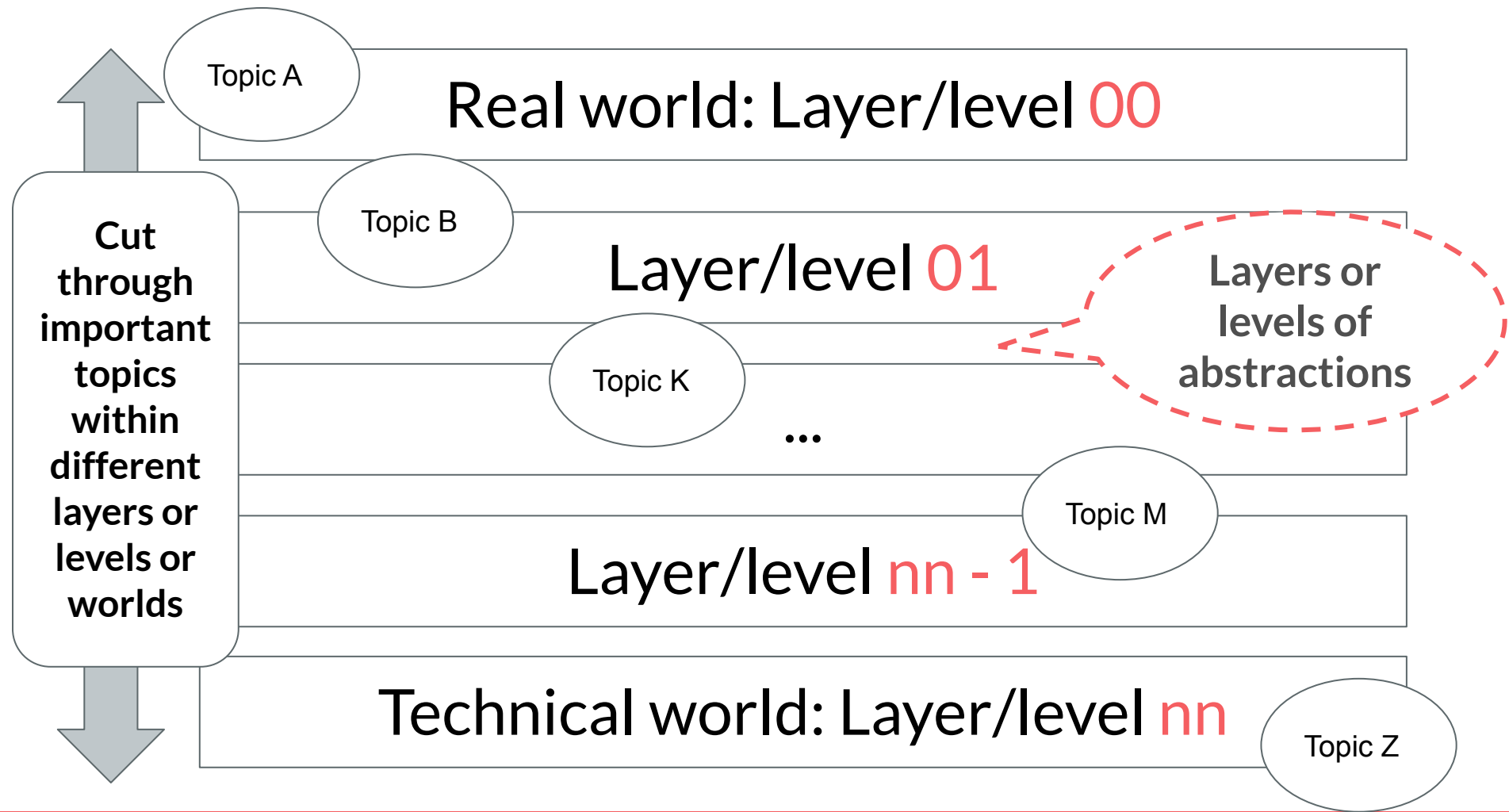
And share some
tips!



Instead I'll try to share
perspectives, on how
things sit together in my
experience!

Share approaches and
how I try to go around
things!

**Share methods and
maybe some of my
work! In additional
to all other useful
resources!**



Why “nn” things?

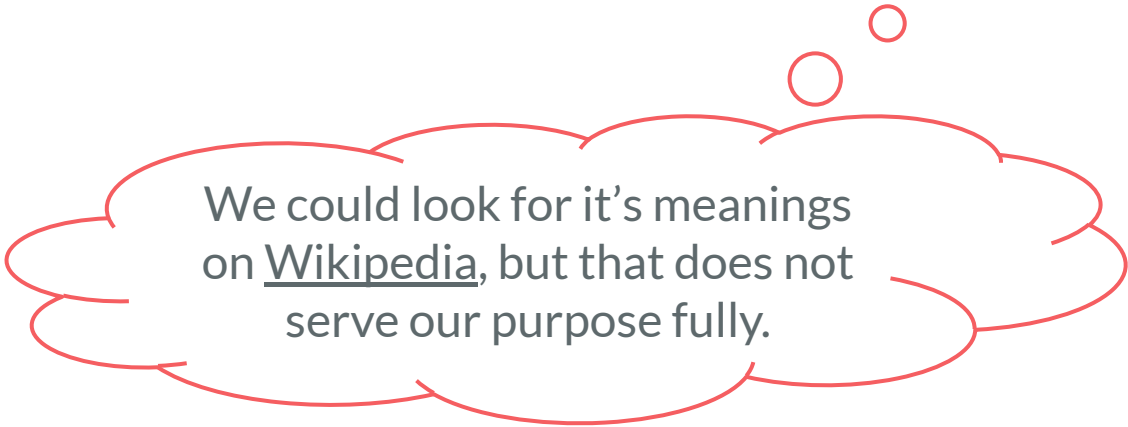
Each time we
know “n” things...

We find that there
are “n” plus “more”
things to know

It can give rise to
something like
“imposter syndrome”
or just “overwhelm” us!

**Only way forward
is to remove such
impediments!**

What is AI / ML / DL ?



We could look for it's meanings on Wikipedia, but that does not serve our purpose fully.

Mish-mash of many topics,
techniques, methods! They
can overlap yet they can be
different and separate

Which kind of
model to build
maybe an incorrect
question!

Understanding
data over model
building, see
talks

Multiple meanings,
overlapping meanings,
misconceptions and
hype

Do your own
research! Don't
just follow the
hype!

Interpretable and
explainable models
over black box
models!

Levels of understanding, abstractions, and views

Real world

People, groups, communities, non-profit organisations

Business

Products, services, costs, profits, regulations, etc....

**The “real world” does
not see many things,
just what’s on the
surface!**

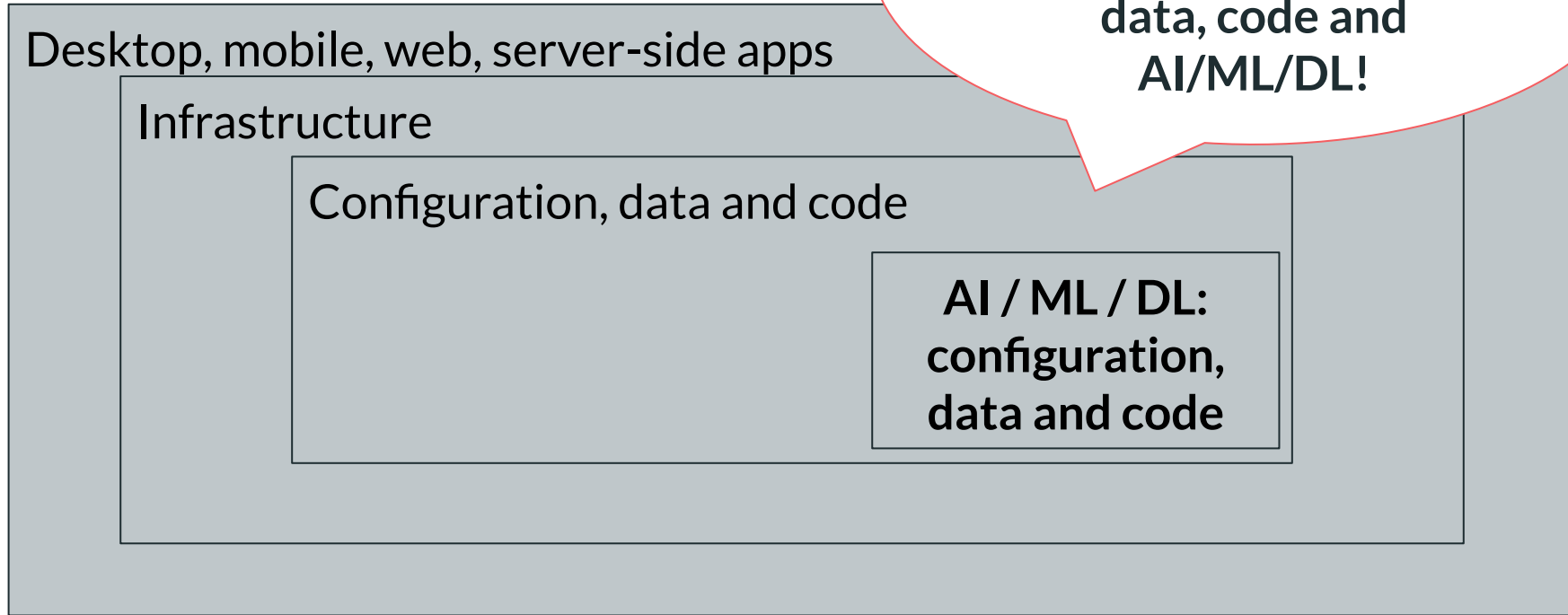
Business

Products, services, costs, profits, regulations, etc....

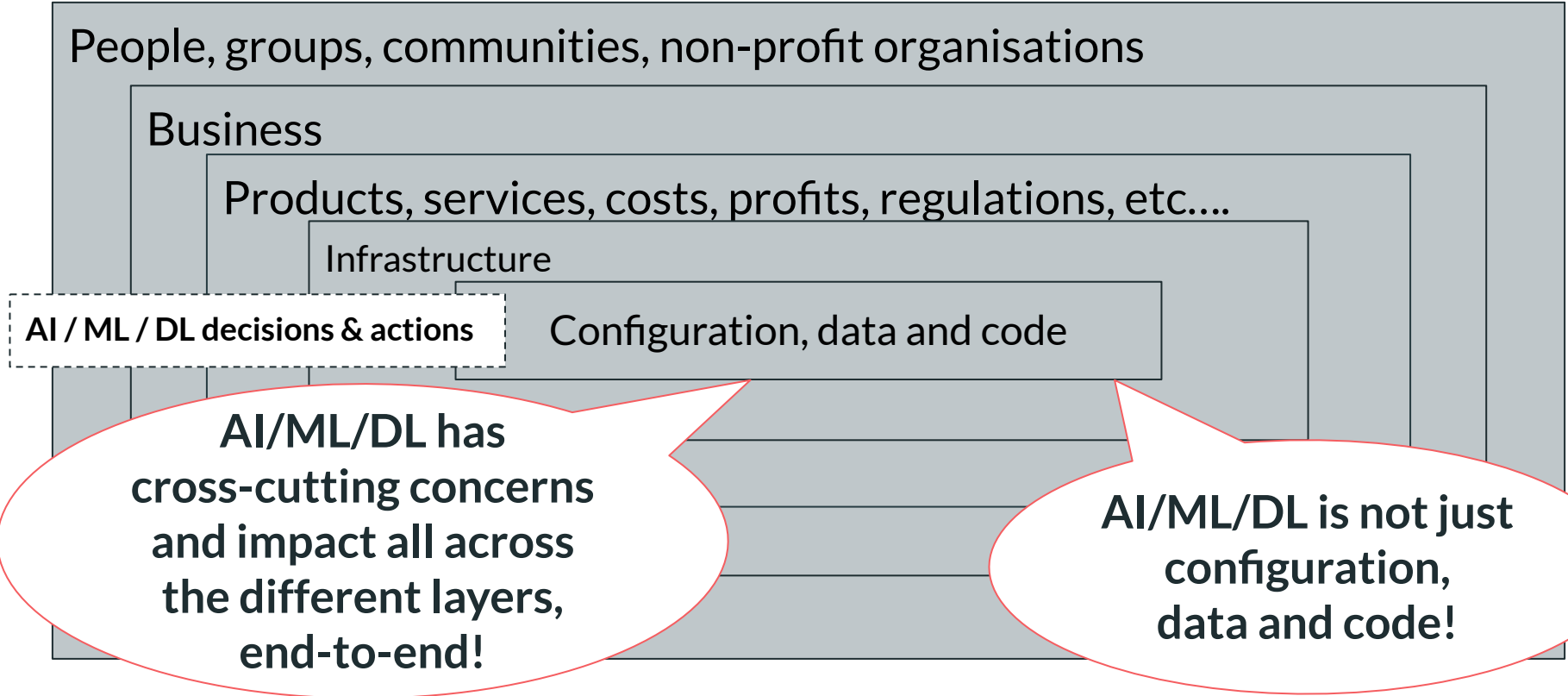
Desktop, mobile, web, server-side apps

**Business world may
not have the full
insight on how
things sit together!**

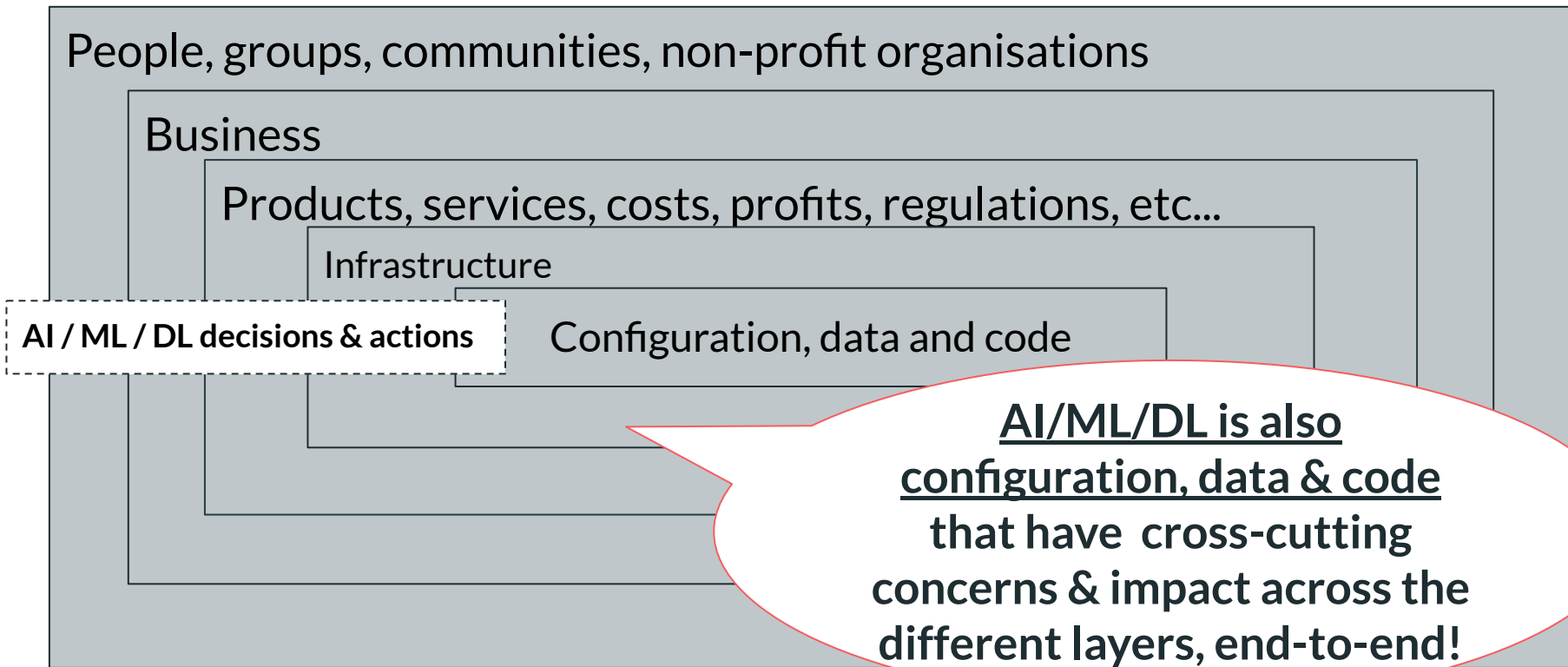
Technical



In reality



In reality



And then there are other realities!

How it all started for
me?

Timeline

- January to November 2018: collecting links
- 2018 December: creation of [awesome-ai-ml-dl](#)
- Early 2019: [Better NLP](#) library
- 2019: collaboration with [Virgilio](#) (reviewing guides)
- May 2019: talk on data, see [talks slide](#)
- Mid 2019: another [Better NLP](#) presentation
- Rest of 2019: blogs on AI/ML/DL and example projects
- Late 2019, early 2020: online Data/ML competitions
- 2020: more links added to [awesome-ai-ml-dl](#)

Thanks to Yolande Poirier
for posting all those
AI/ML/DL links on @java
between 2016 and 2018

Thanks to the
Meet-a-Mentor initiative by
LJC during 2018-19, for
holding the ML Study
Group!

Thanks to Yaz for the
Tensorflow meetups in
London, UK during 2018 and
2019

**And many others in this
journey...**


How do I learn?

Top-down, outside in learning, see Rachel Thomas' talk (fast.ai)

Mind-maps, gathering reading lists, organising them, categorising them

Nodes of Knowledge (see Appendix, previous talks)

Automatic chunking, Learning to Learn course by Coursera



List building: helps
know the landscape
and understand the
context:
birds-eye-view


List building:
helps prioritise,
order, remove
or add topics
you like, or
don't like or
don't know

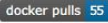
AI/ML/DL git repo

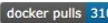

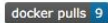
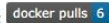
<https://github.com/neomatrix369/awesome-ai-ml-dl>


README.md


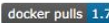
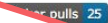
Awesome AI-ML-DL


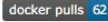
 awesome License CC BY-SA 4.0

Better NLP: 

NLP Java:  | NLP Clojure:  | NLP Kotlin:  | NLP Scala: 

NLP using DL4J (cuda) 

Dataiku DSS:  | Grakn:  | Jupyter-

MLPMNist using DL4J:  | Zeppelin: 

Awesome Artificial Intelligence, Machine Learning and Deep Learning as we learn from the best of awesome resources of such topics.

This repo is dedicated to engineers, developers, data scientists and all other professionals in the field of AI, ML, DL and related sciences. To make learning interesting and to create a necessary material. Please contribute, watch, star, fork and share the repo with your friends.


Watching the repo will keep you posted of all the changes (commits) that go into this repo.

Also, please [SPONSOR us](#), [find out how-to!](#)

- [General](#)
- [Artificial Intelligence](#)
- [Automation](#)
- [Competitions](#)
- [Courses](#)

No packages published
[Publish your first package](#)

Contributors 15



+ 4 contributors

Please *star*, watch,
fork and share the
repo, as it motivates
me to continue with
my work

AI/ML/DL highlights

My first post on AI/ML/DL!

Courses

ML on
Code

Data Science

Mathematics, Statistics, Probability &
Probabilistic programming

Neural
Networks

NLP

Graphs

Genetic Algorithms

Machine Learning

JavaScript

Natural Language Processing

Competition

Cloud, DevOps, Infra

Visualisation

Time-series / anomaly detection

Data

AI/ML/DL & Java highlights

[Classifiers and Decisions trees](#)

[Data Science](#)

[Deep Learning](#)

[Neural Networks](#)

[Tools, libraries
& resources](#)

[Genetic Algorithms](#)

[Machine Learning](#)

[Natural Language Processing](#)

[JSRs](#)

[Clojure resources](#)

[Scala resources](#)

[Java related projects and
technologies](#)

AI/ML/DL artefacts

Cheatsheets

Notebooks

Presentations

Examples

Virgilio

Guides

Docker
containers


Study Notes

Things to know list

Learning by example

See Appendix: Learning by example

- DL4J example
- NLP example
- Jupyter Notebook example
- Apache Zeppelin example
- grql-to-english, english-to-grql
- grCUDA example
- grPython examples



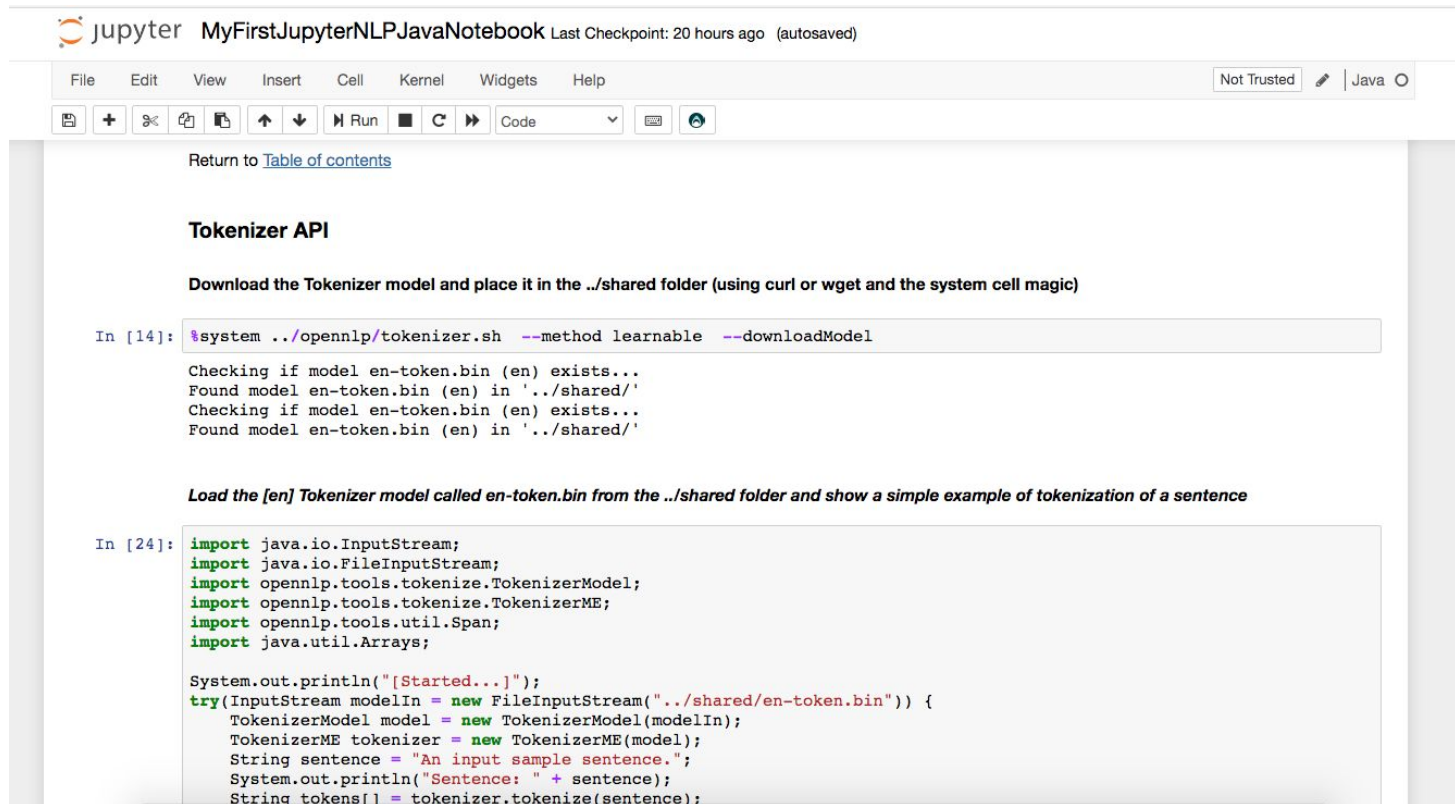
12+ months
worth of
coding work

Demo

Running Java inside a Jupyter Notebook

- Blog post
- Github repo nlp-java-jvm-example

Running Java inside a Jupyter Notebook



The screenshot shows a Jupyter Notebook titled "MyFirstJupyterNLPJavaNotebook" with a "Last Checkpoint: 20 hours ago (autosaved)" status. The interface includes a menu bar (File, Edit, View, Insert, Cell, Kernel, Widgets, Help) and a toolbar with icons for saving, adding cells, and running code. The notebook content is as follows:

Return to [Table of contents](#)

Tokenizer API

Download the Tokenizer model and place it in the `../shared` folder (using curl or wget and the system cell magic)

```
In [14]: %system ../opennlp/tokenizer.sh --method learnable --downloadModel
```

Checking if model en-token.bin (en) exists...
Found model en-token.bin (en) in '../shared/'
Checking if model en-token.bin (en) exists...
Found model en-token.bin (en) in '../shared/'

Load the [en] Tokenizer model called en-token.bin from the ../shared folder and show a simple example of tokenization of a sentence

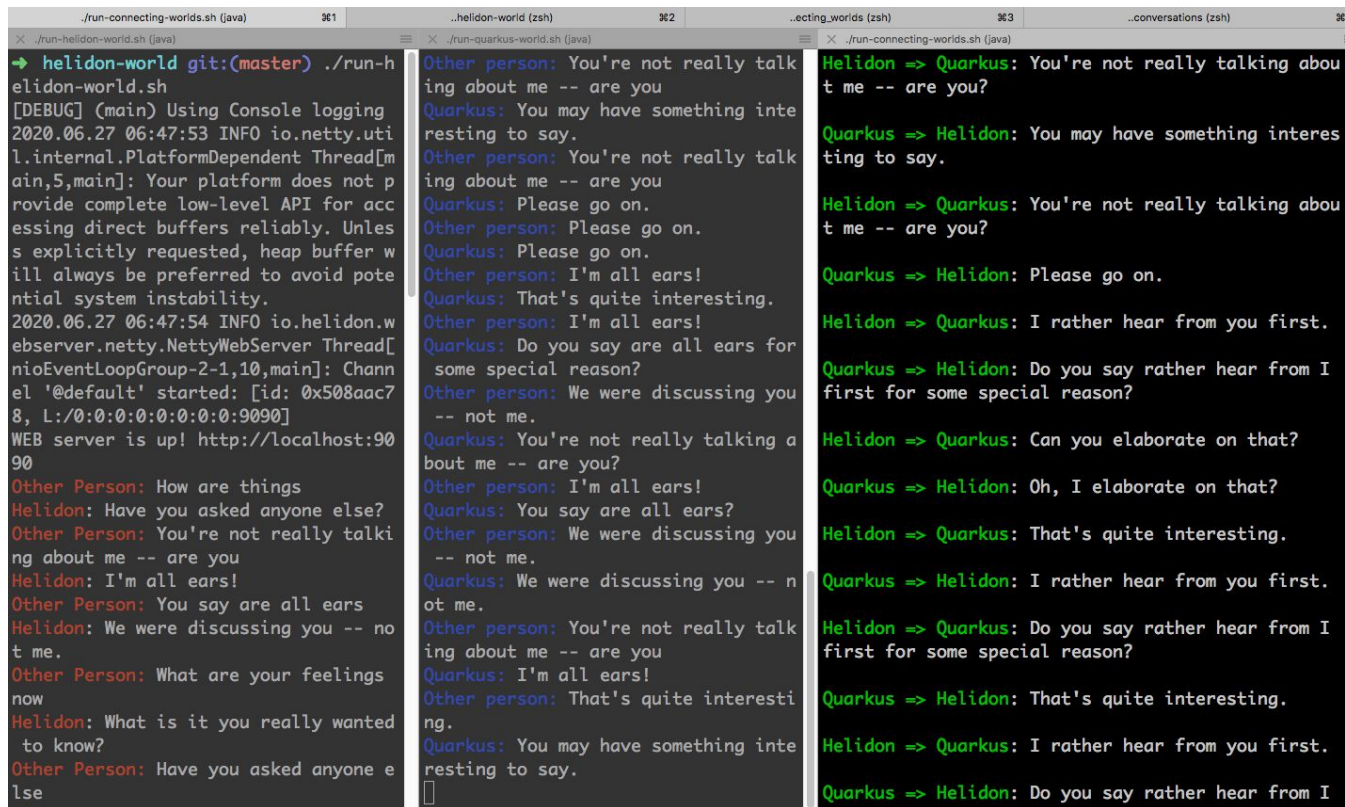
```
In [24]: import java.io.InputStream;
import java.io.FileInputStream;
import opennlp.tools.tokenize.TokenizerModel;
import opennlp.tools.tokenize.TokenizerME;
import opennlp.tools.util.Span;
import java.util.Arrays;

System.out.println("[Started...]");
try(InputStream modelIn = new FileInputStream("../shared/en-token.bin")) {
    TokenizerModel model = new TokenizerModel(modelIn);
    TokenizerME tokenizer = new TokenizerME(model);
    String sentence = "An input sample sentence.";
    System.out.println("Sentence: " + sentence);
    String tokens[] = tokenizer.tokenize(sentence);
```

Making chatbot have a conversation

- Github repo:
<https://github.com/neomatrix369/chatbot-conversations>
- Questions
 - Is this a real conversation?
 - Would this pass the Turing test?
 - What is good about this example/demo?
- Puzzles to solve
 - What are the different things that can be improved?
 - What new ideas come to mind when you see this?

Making chatbot have a conversation

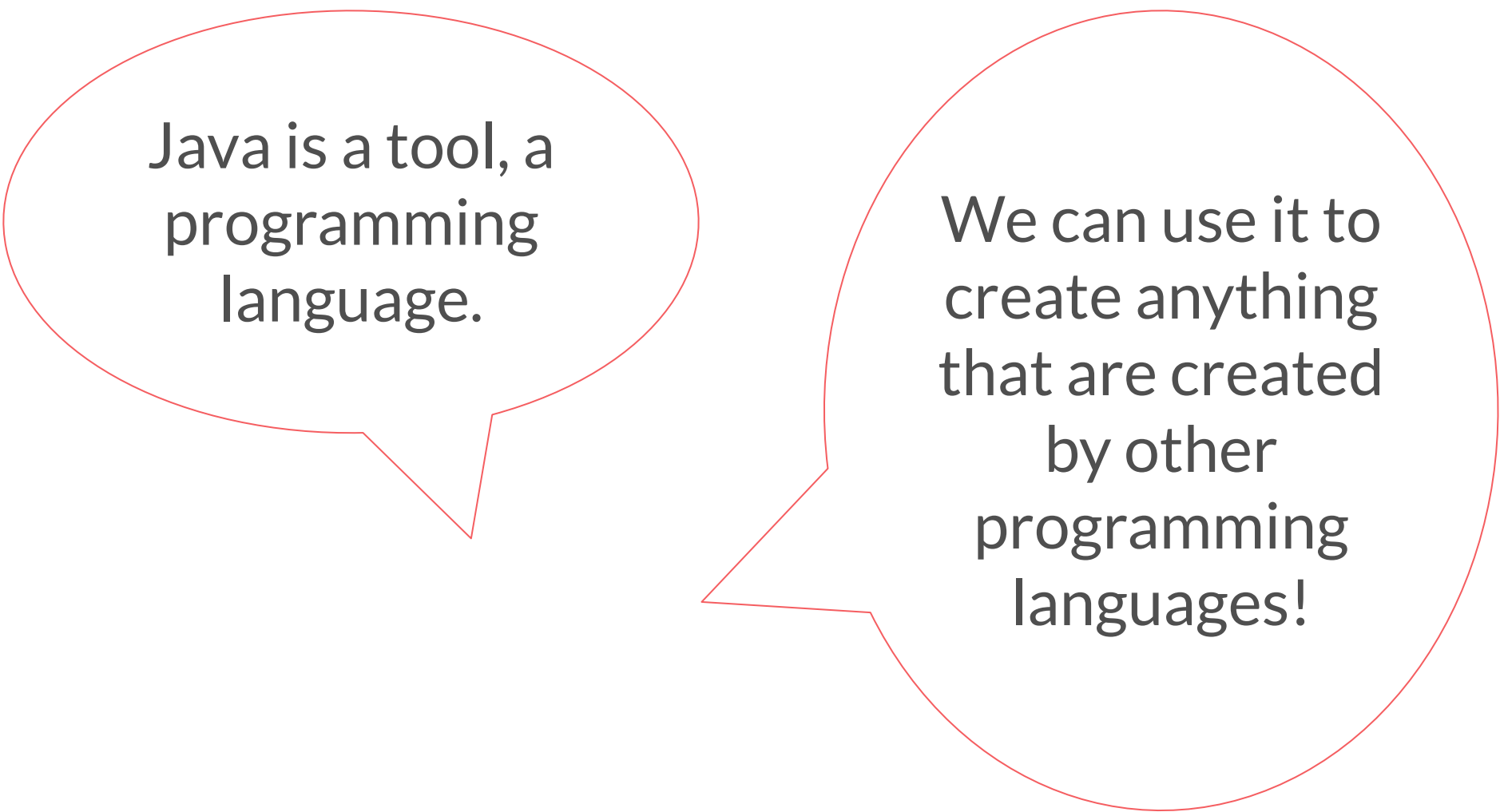


The screenshot displays a terminal window with four panels, each showing a different perspective of a chatbot conversation. The panels are titled: `./run-connecting-worlds.sh (java)`, `./helidon-world (zsh)`, `./connecting-worlds (zsh)`, and `./conversations (zsh)`. The first panel shows the Java application's internal logs and the start of the chatbot. The subsequent panels show the dialogue from the perspectives of the two chatbot characters, Helidon and Quarkus, as they interact with each other.

```
./run-connecting-worlds.sh (java) 361
X ./run-helidon-world.sh (java)
helidon-world git:(master) ./run-helidon-world.sh
[DEBUG] (main) Using Console logging
2020.06.27 06:47:53 INFO io.netty.util.internal.PlatformDependent Thread[m
ain,5,main]: Your platform does not provide complete low-level API for accessing direct buffers reliably. Unless explicitly requested, heap buffer will always be preferred to avoid potential system instability.
2020.06.27 06:47:54 INFO io.helidon.webserver.netty.NettyWebServer Thread[nioEventLoopGroup-2-1,10,main]: Channel 'default' started: [id: 0x508aac78, L:/0:0:0:0:0:0:0:0:9090]
WEB server is up! http://localhost:9090
Other Person: How are things
Helidon: Have you asked anyone else?
Other Person: You're not really talking about me -- are you
Helidon: I'm all ears!
Other Person: You say are all ears
Helidon: We were discussing you -- not me.
Other Person: What are your feelings now
Helidon: What is it you really wanted to know?
Other Person: Have you asked anyone else

./helidon-world (zsh) 362
Other person: You're not really talking about me -- are you
Quarkus: You may have something interesting to say.
Other person: You're not really talking about me -- are you
Quarkus: Please go on.
Other person: Please go on.
Quarkus: Please go on.
Other person: I'm all ears!
Quarkus: That's quite interesting.
Other person: I'm all ears!
Quarkus: Do you say are all ears for some special reason?
Other person: We were discussing you -- not me.
Quarkus: You're not really talking about me -- are you?
Other person: I'm all ears!
Quarkus: You say are all ears?
Other person: We were discussing you -- not me.
Quarkus: We were discussing you -- not me.
Other person: You're not really talking about me -- are you
Quarkus: I'm all ears!
Other person: That's quite interesting.
Quarkus: You may have something interesting to say.

./connecting-worlds (zsh) 363
Helidon => Quarkus: You're not really talking about me -- are you?
Quarkus => Helidon: You may have something interesting to say.
Helidon => Quarkus: You're not really talking about me -- are you?
Quarkus => Helidon: Please go on.
Helidon => Quarkus: I rather hear from you first.
Quarkus => Helidon: Do you say rather hear from I first for some special reason?
Helidon => Quarkus: Can you elaborate on that?
Quarkus => Helidon: Oh, I elaborate on that?
Helidon => Quarkus: That's quite interesting.
Quarkus => Helidon: I rather hear from you first.
Helidon => Quarkus: Do you say rather hear from I first for some special reason?
Quarkus => Helidon: That's quite interesting.
Helidon => Quarkus: I rather hear from you first.
Quarkus => Helidon: Do you say rather hear from I
```



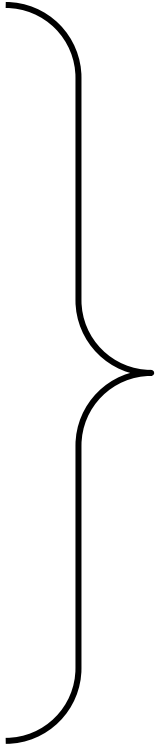
Java is a tool, a
programming
language.

We can use it to
create anything
that are created
by other
programming
languages!

Resources

Learning websites

- Awesome AI/ML/DL
- Better NLP
- Awesome Graal
- Virgilio | GitHub
- MadeWithML



All you need to know, pick and choose, make your own collections / checklists / playlists of resources to study and practice from.

More resources

See Appendix section

Summary

- My journey and how I learnt the topics
- We get better results when we organise ourselves
- Create something simple from scratch, similar to our demo and other examples
- A simple idea can go very far, inspire others with creative solutions

Summary

- Java is versatile and can be used to create amazing things just like others are creating
- We didn't cover even more number of Java and AI/ML/DL related topics, libraries framework as the resources shared here hopefully will lead up to them
- No need to feel overwhelmed and let's try to avoid "imposter syndrome"

Summary

The pathway to mastery or championing a subject means we must be different and take different approaches, in addition to the ones that have already been taken!

Thank you!

**Martin Toshev and
Martin Patsov!**



**Use Hashtag:
#jOnConf**

Questions & feedback

Use Hashtag:
#jOnConf

Please share your questions
and feedback at

@theNeomatrix369

or on the video stream

Social media links

- twitter: [@theNeomatrix369](#)
- medium: <https://medium.com/@neomatrix369>
- github: <https://github.com/neomatrix369/>
- linkedin:
<https://uk.linkedin.com/pub/mani-sarkar/71/a77/39b>
- slideshare: <https://www.slideshare.net/neomatrix369/>
- youtube: [My Channel](#) | [Videos playlist](#)

Use Hashtag:
#jOnConf

Keep in mind...



It's your turn next to share and inspire!!!

Appendix

Freebies!

Get \$500 worth of free cloud
credits on Oracle Cloud

People doing some great work with AI & Java

Eyal Wirsansky, Zoran Sevarac,
Suyash Joshi, Adam Pocock,
Johan Vos and many more...

(please share more names and examples with me so I can add them here)

Accelerated Processing

Java on the GPU by Mitia

Plenty of resources on  **NVIDIA**®

Nvidia's [Developer site](#) | [Community](#) | [Research](#) | [Blog site](#)

Parallel processing

GNU
Parallel

<https://www.gnu.org/software/parallel/>

Parallel / Async
programming

[Talk by Venkat Subramanian](#)

ASYNC
Reactive programming

<https://community.oracle.com/docs/DOC-1006738>

Java and AI/ML/DL

Machine Learning Best Practices

o

Top 5 machine learning libraries for Java

o

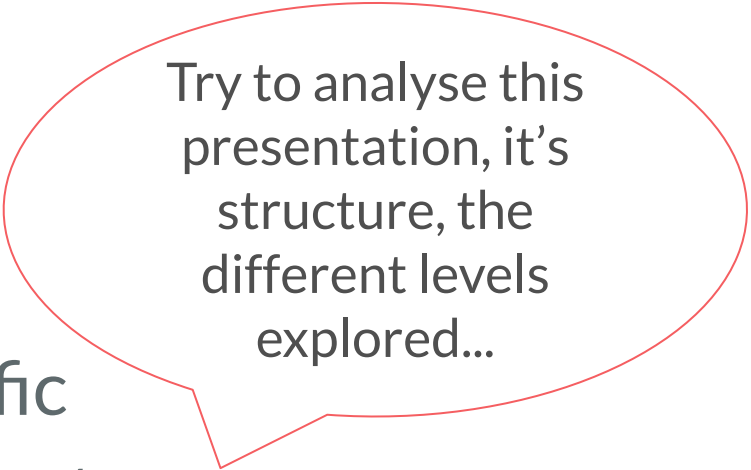
10 Popular Java Machine Learning Tools & Libraries

o

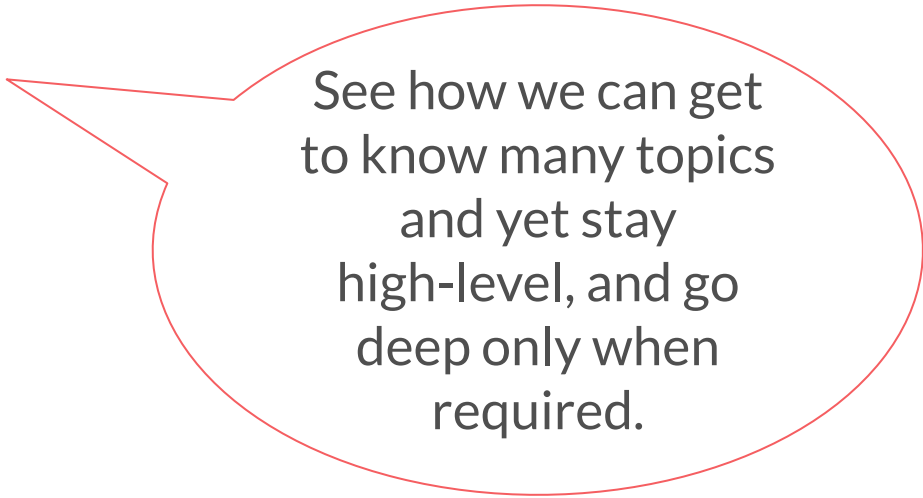
What are machine learning libraries in Java?

Levels of abstractions

- Higher to lower
 - From generic to more specific
- Always know the level of abstract
 - when reading
 - when writing
 - when speaking



Try to analyse this presentation, it's structure, the different levels explored...



See how we can get to know many topics and yet stay high-level, and go deep only when required.

Book references

- How to think like a Scientist!
- Deep work book
- Getting Things Done: Book | Free Resources

Previous talks

- Java N.n: What to know? How to learn?
- Do we know our data as well as our tools?
- Naturally, getting productive, my journey with Grakn and Graql

Create something simple from scratch

- even if it's as simple or silly idea - like the demos i have shown
- *(it can also be putting existing components together from scratch)*
- this can have a massive impact on us and our communities

Learning by example

DL4J example

- Github
- Blog post

NLP examples

- Example 1
 - Github
 - Blog post
- Example 2
 - Blog post
- Example 3
 - Blog post
- Better NLP

Jupyter Notebook example

- Example 1
 - Github
 - Blog: Exploring NLP concepts using Apache OpenNLP inside a Jupyter notebook
- Example 2
 - Blog post
- Example 3
 - Github
 - Blog post

Apache Zeppelin example

- Github
- Blog posts
 - Apache Zeppelin: stairway to notes* haven!
 - Running Apache Zeppelin on Oracle Cloud Infrastructure

grql-to-english, english-to-grql example

- Presentation
- Github

grCuda example

- Blog posts

- [grCUDA: A Polyglot Language Binding for CUDA in GraalVM.](#)
NVIDIA Developer Blog, November 2019.
- [grCUDA: A Polyglot Language Binding.](#) Presentation at Oracle
CodeOne 2019, September 2019.
- [Simplifying GPU Access.](#) Presentation at NVIDIA GTC 2020, March
2020
- [Optimizing Machine Learning Performance at Netsuite with](#)
GraalVM and NVIDIA GPUs

- [Github](#)

graalPython examples

- Blog posts
 - [Introduction to the Python implementation for GraalVM](#)
 - [Moving from Jython to GraalVM](#)
 - [Running Python on GraalVM](#)
- [Github](#)