

As Presented @UtrechtJUG

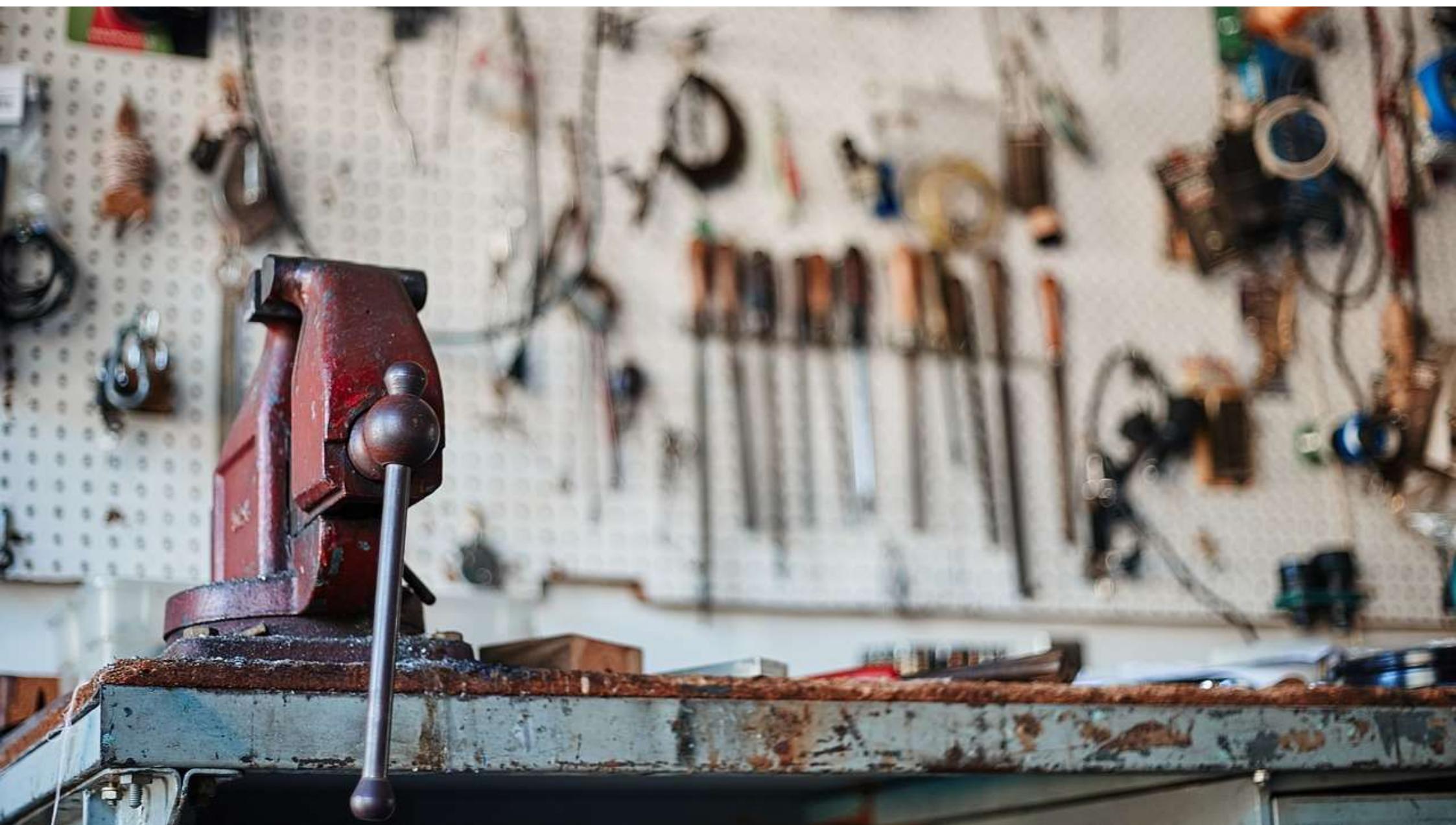
Code Examples: <https://tomcools.be/talks/learningthroughtinkering-utrechtjug/>

# Learning Through Tinkering

Tom Cools  
@TCoolsIT

[www.tomcools.be](http://www.tomcools.be)

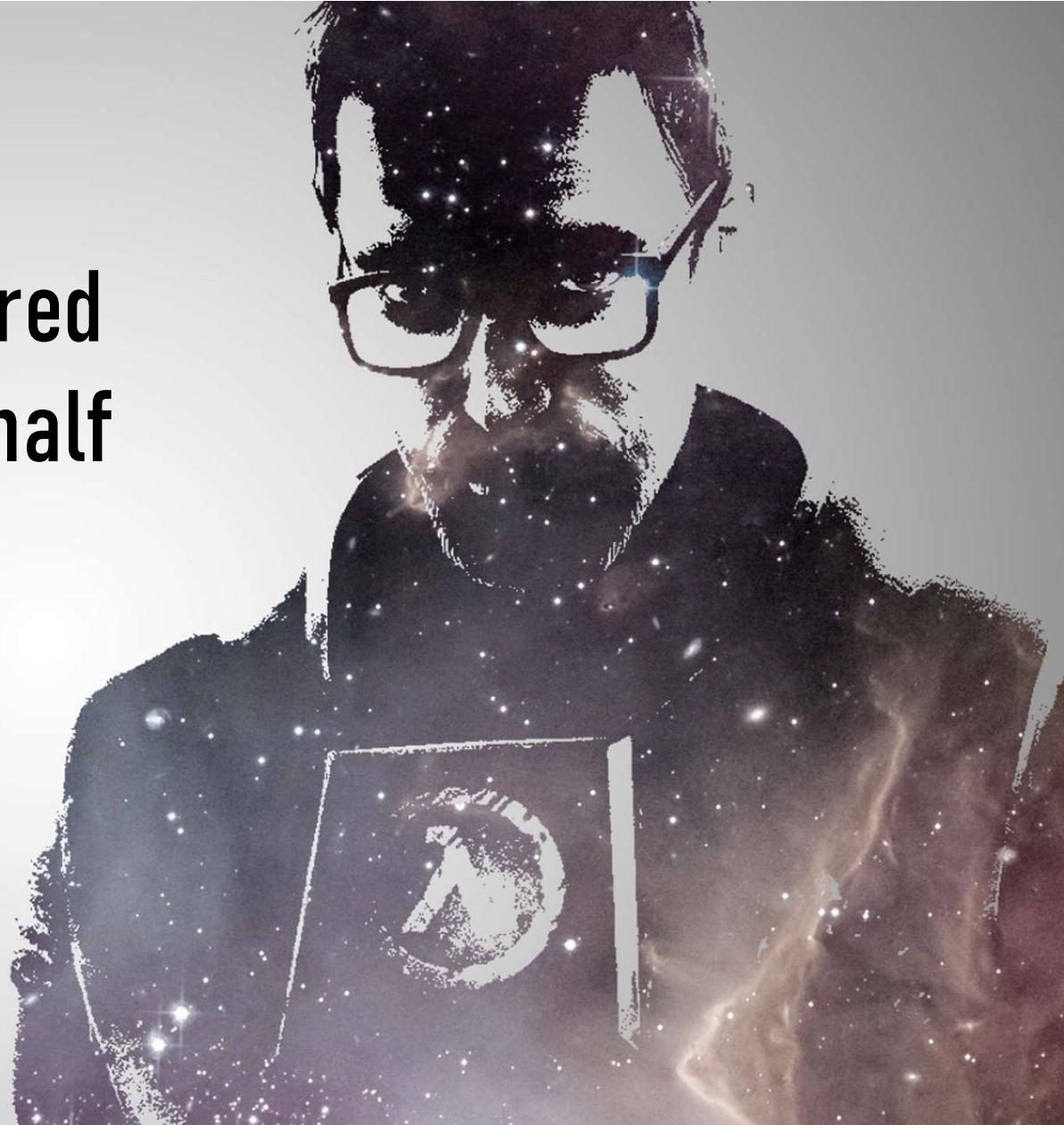




# A hype today could be a standard tomorrow

- Quarkus (March 7, 2019)
- GraalVM 1.0-RC2 (June 5, 2018)
- Spring Webflux (September 28, 2017)
- Kotlin (February 15, 2016)
- Kubernetes (July 21, 2015)
- Spring Boot (April, 2014)
- Docker (March 13, 2013)

**“Half-Life** is the time required  
for a quantity to reduce to half  
its initial value.”



The **Half-Life** of relevant  
knowledge in the IT-Sector:

**2 to 5 years**



# Learning Through Tinkering

- What should we be learning?
- Apply what you learned... but how?
- Why do we stop before it's finished?
- When should we be tinkering?

# Learning Through Tinkering

- What should we be learning?
- Apply what you learned... but how?
- Why do we stop before it's finished?
- When should we be tinkering?





# Picking a new tool to learn...

Twitter

Conferences

Technology Radar

Meetups (like this one 😊)

**Can't learn this! 😞**

**Can learn this! 😊**

**What you know**

Zone of Proximal Development  
Lev Vygotsky

**“Technical implementations die faster  
than the concepts they are based on.”**



# Learning Through Tinkering

- What should we be learning?
- Apply what you learned... but how?
- Why do we stop before it's finished?
- When should we be tinkering?

Let's learn through Tinkering!

“Experimenting with ideas/frameworks  
in an integrated environment  
to fully understand their capabilities”

# Limit Yourself





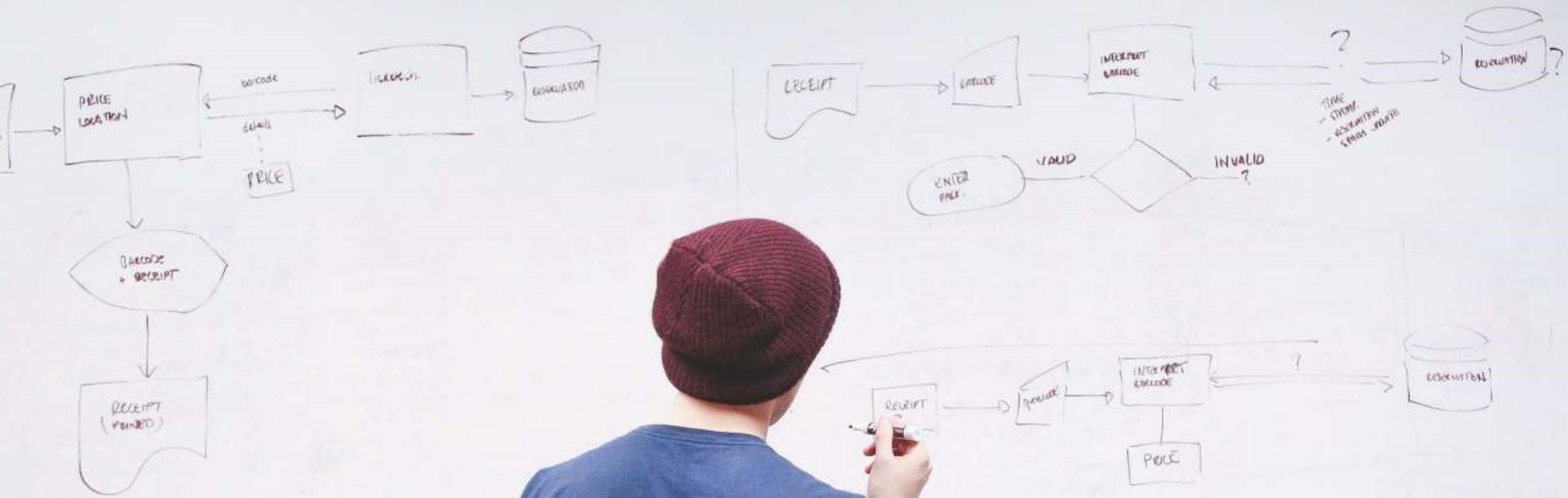


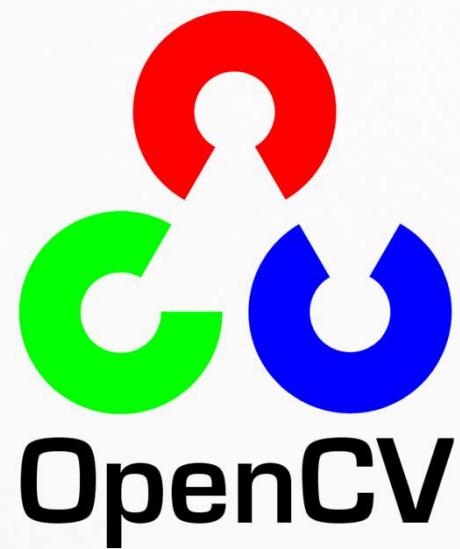
# Limit Yourself



# Automate Whatever

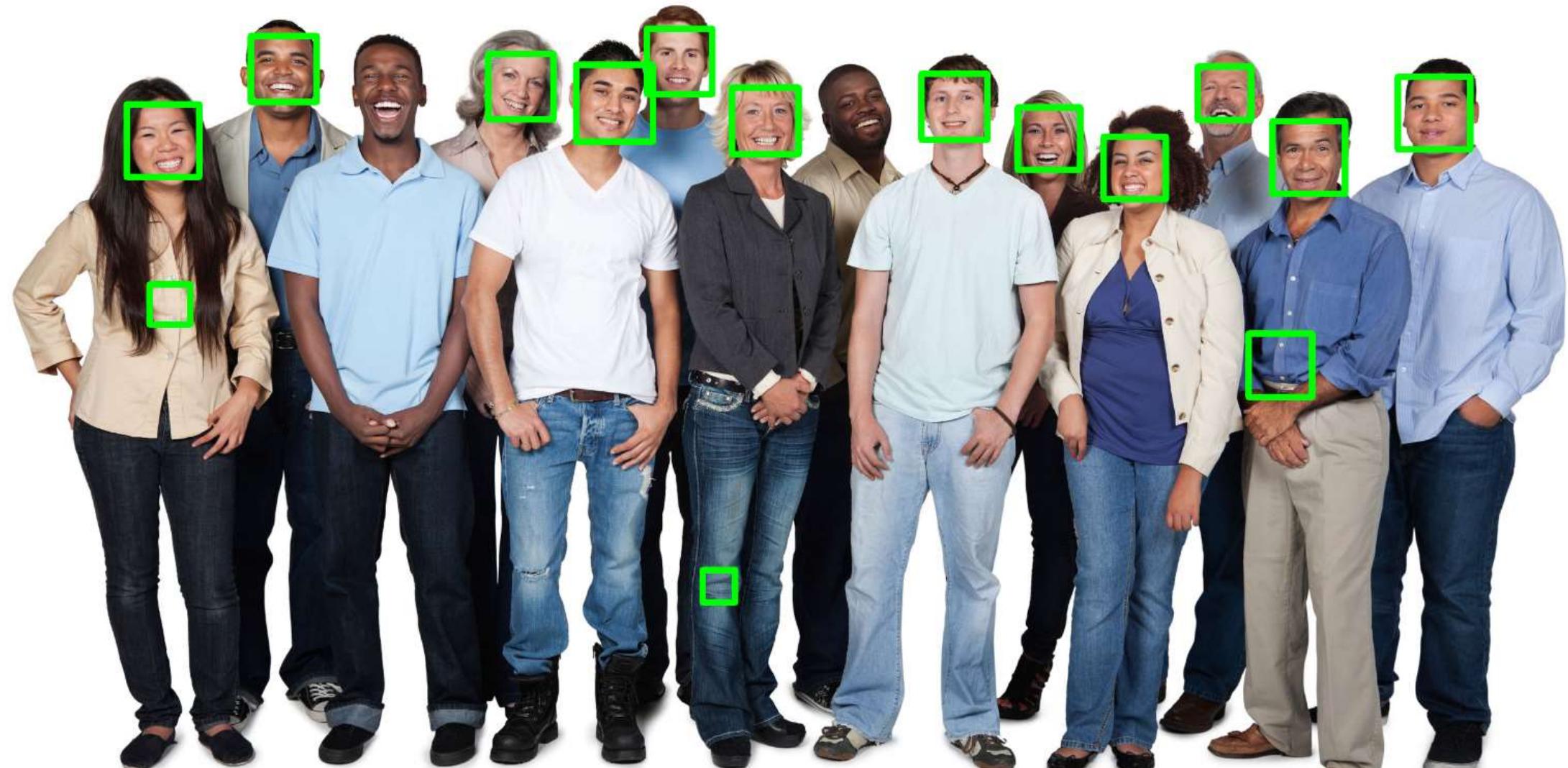






Microsoft Cognitive  
Services  
Face API



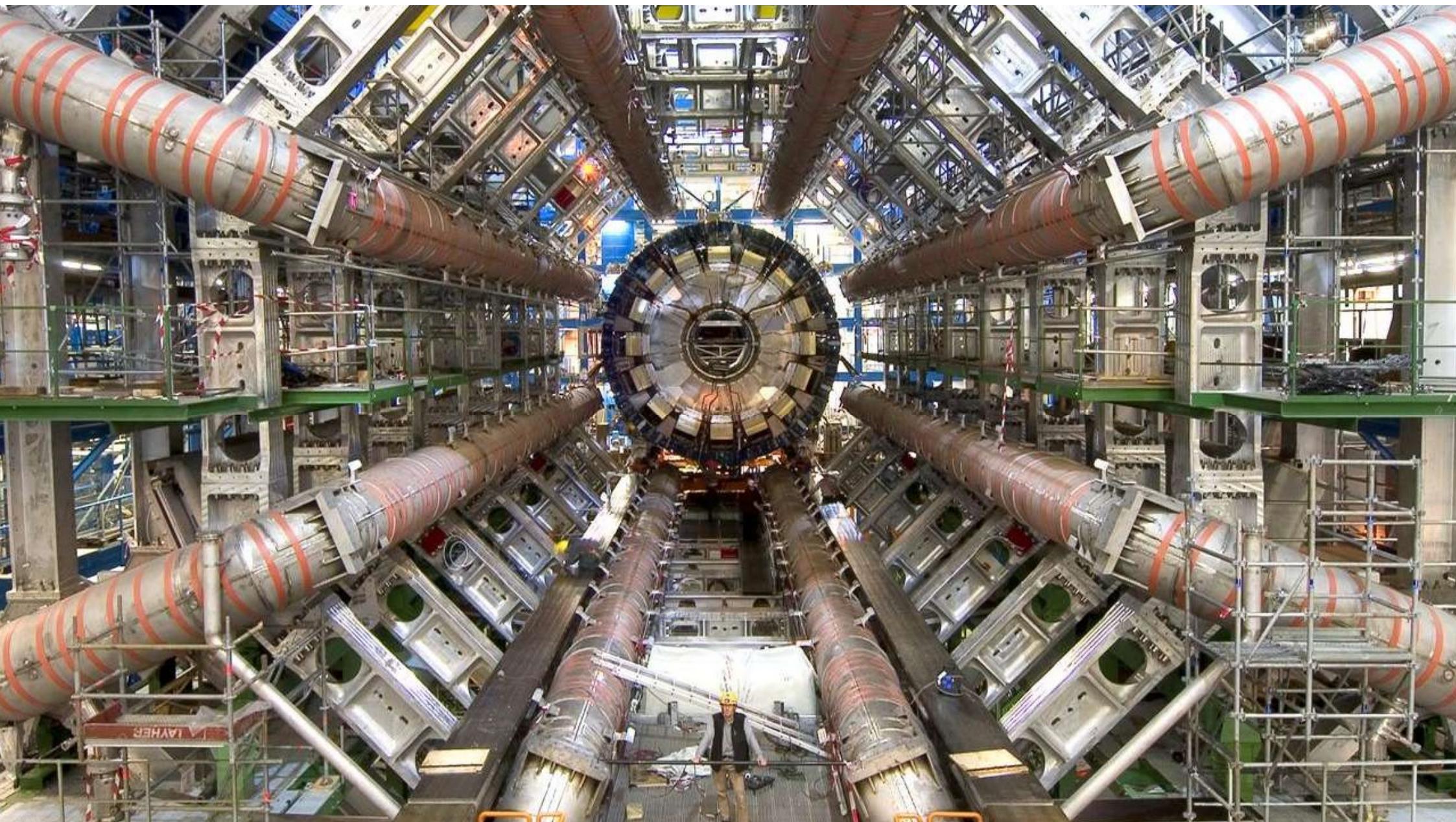


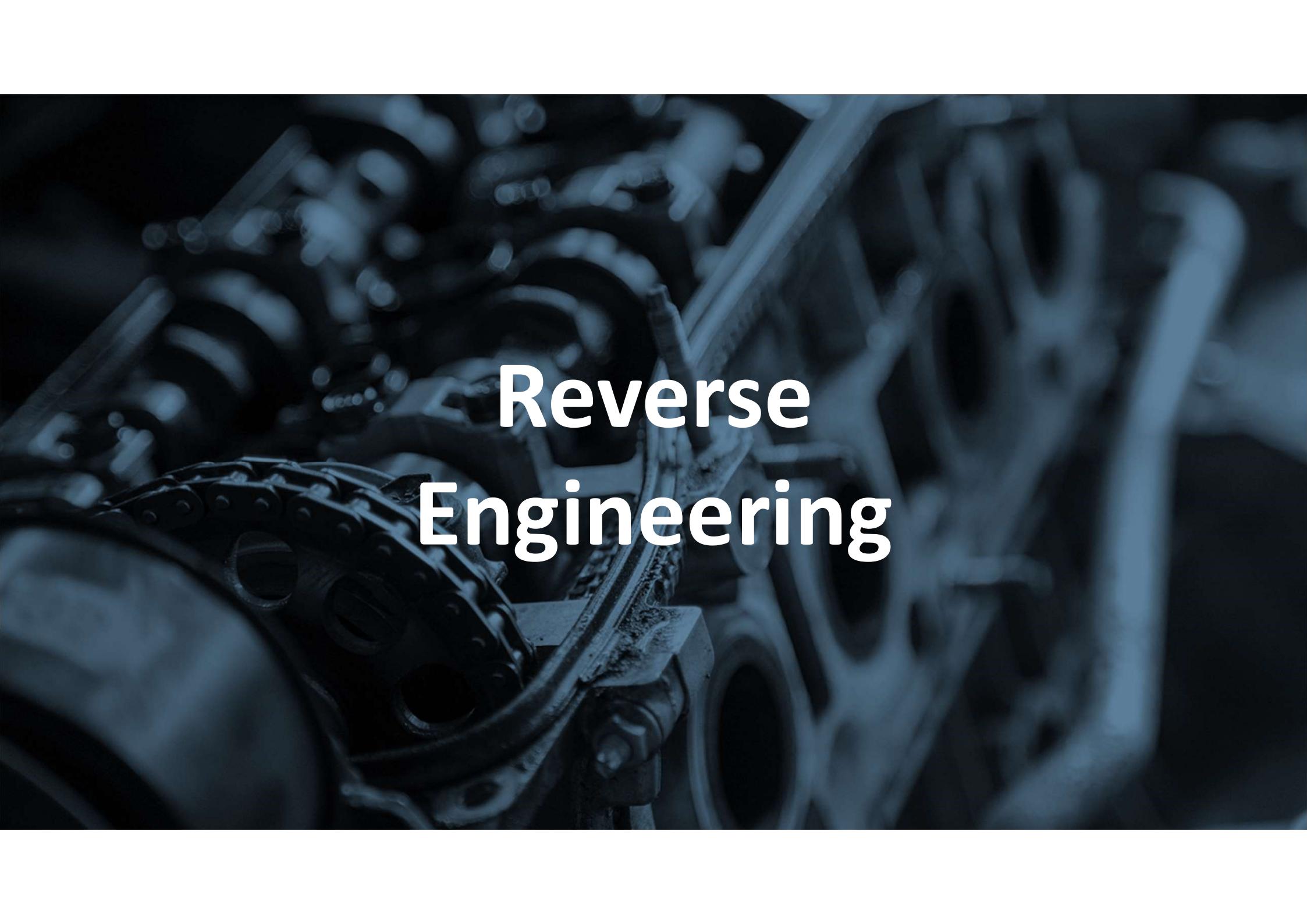
# Automate Whatever



# Replicate







# Reverse Engineering



<https://www.reddit.com/r/pokemongodev/>

Path: /plfe/rpc

Method: POST

Body: □□? ? ? ? ? ? ? ? S"□□

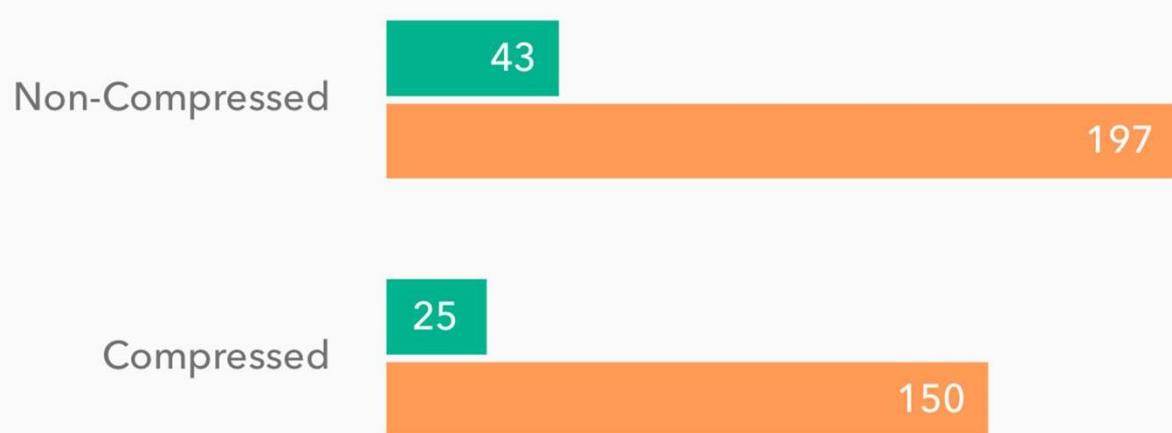
# No JSON!?!

# No, it's Protocol Buffers!

Source:

[https://www.reddit.com/r/pokemongodev/comments/4trjum/pok%C3%A9mon\\_go\\_development\\_archive\\_works\\_in\\_progress/](https://www.reddit.com/r/pokemongodev/comments/4trjum/pok%C3%A9mon_go_development_archive_works_in_progress/)

## Java to Java Communication



### References

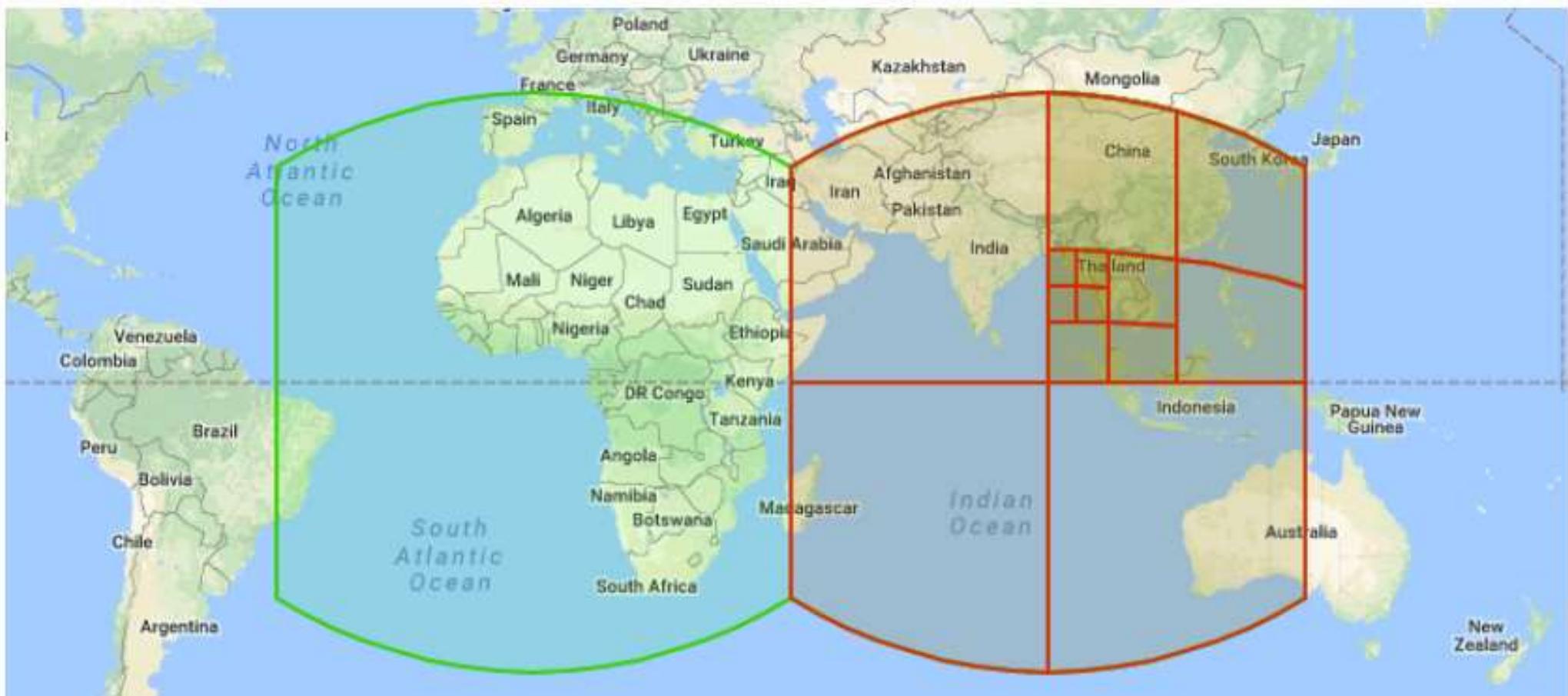
- Protobuf Time (ms)
- JSON Time (ms)

Source: <https://auth0.com/blog/beating-json-performance-with-protobuf/>

# S2 Geometry

From: <http://s2geometry.io/>

Welcome to the S2 Geometry Library!

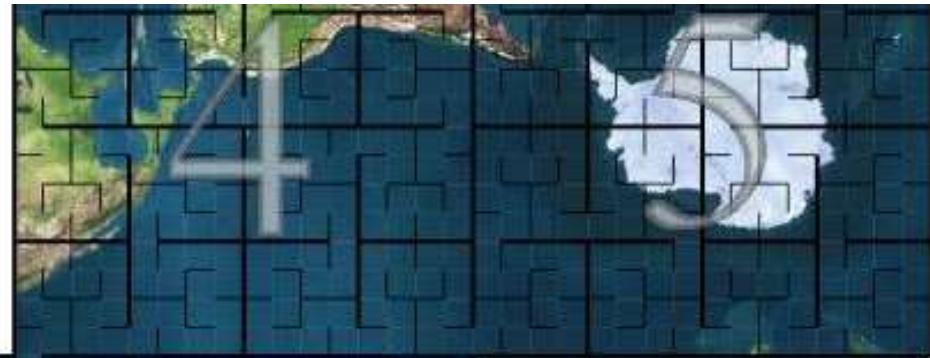
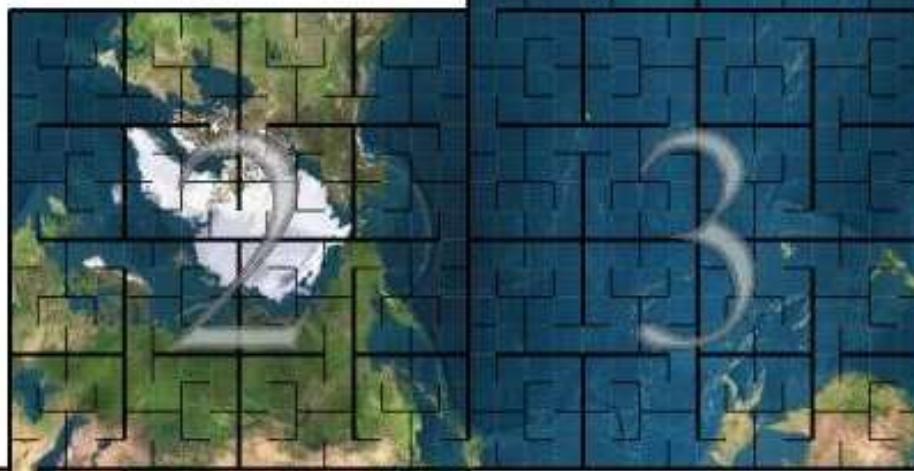


Location based on Latitude and Longitude

This office:

Latitude: 52.026676

Longitude: 5.084564



From: [http://s2geometry.io/devguide/s2cell\\_hierarchy](http://s2geometry.io/devguide/s2cell_hierarchy)

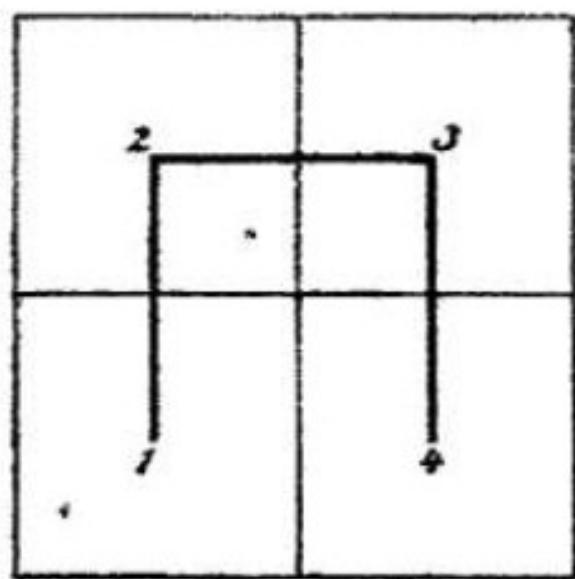


Fig. 1.

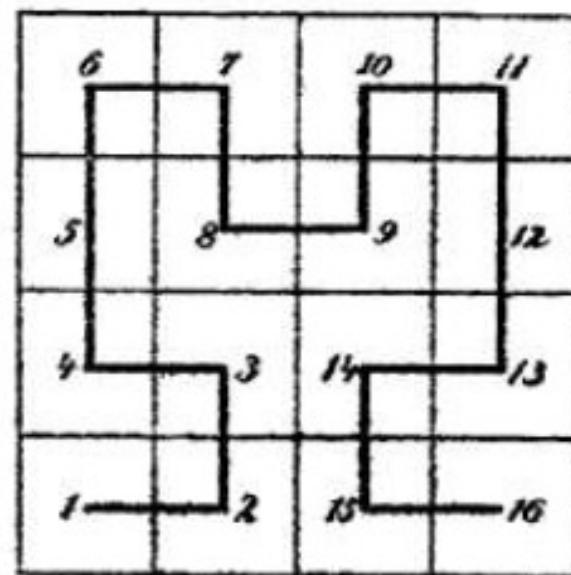
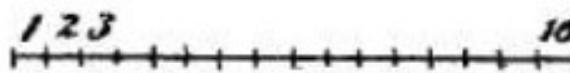


Fig. 2.

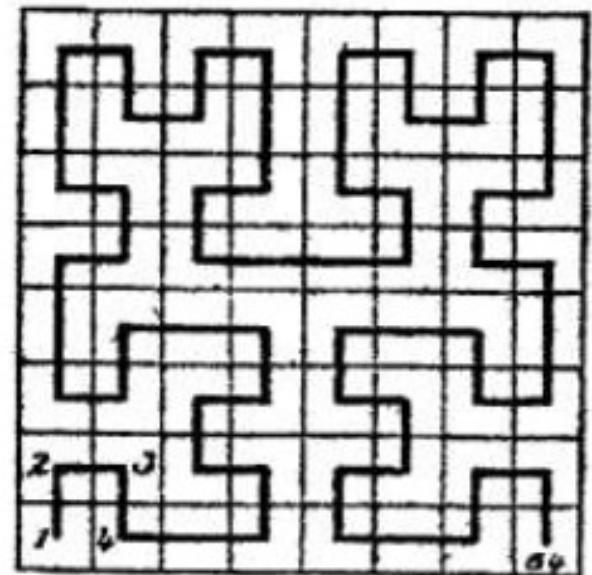


Fig. 3.

From: [http://s2geometry.io/devguide/s2cell\\_hierarchy](http://s2geometry.io/devguide/s2cell_hierarchy)

**100 11 01 1000...0**

L0 (Face) 4

L1 Cell 3

L2 Cell 1

Last 1 = Terminator

If a L0 cell has an area of **85 011 012.19 km<sup>2</sup>**,  
how many levels does it take to get to **~1 cm<sup>2</sup>**?

**30 Levels!**

**30 Levels \* 2 bits = 60 bits**

**3 bits for L0 Faces**

**1 termination bit**

You can map **every cm<sup>2</sup>** on earth in **64 bits!**

**001 00 11 01 00 10 01 01 01 10 10 00 10 00 00 10 11 11 01 10 10 01 10 10 00 10 01 11 00 10 10 1**

**2692B4417B4D1395**

# Replicate



# Challenge









# Challenge





# Learning Through Tinkering

- What should we be learning?
- Apply what you learned... but how?
- Why do we stop before it's finished?
- When should we be tinkering?

Don't like doing it

# Too much new stuff

“Learning is connecting new information to existing knowledge”

*Gerhard Roth, Neuroscientist*

# Misplaced Competition

Seeing  
"Unpaid" Encouraged  
Overtime

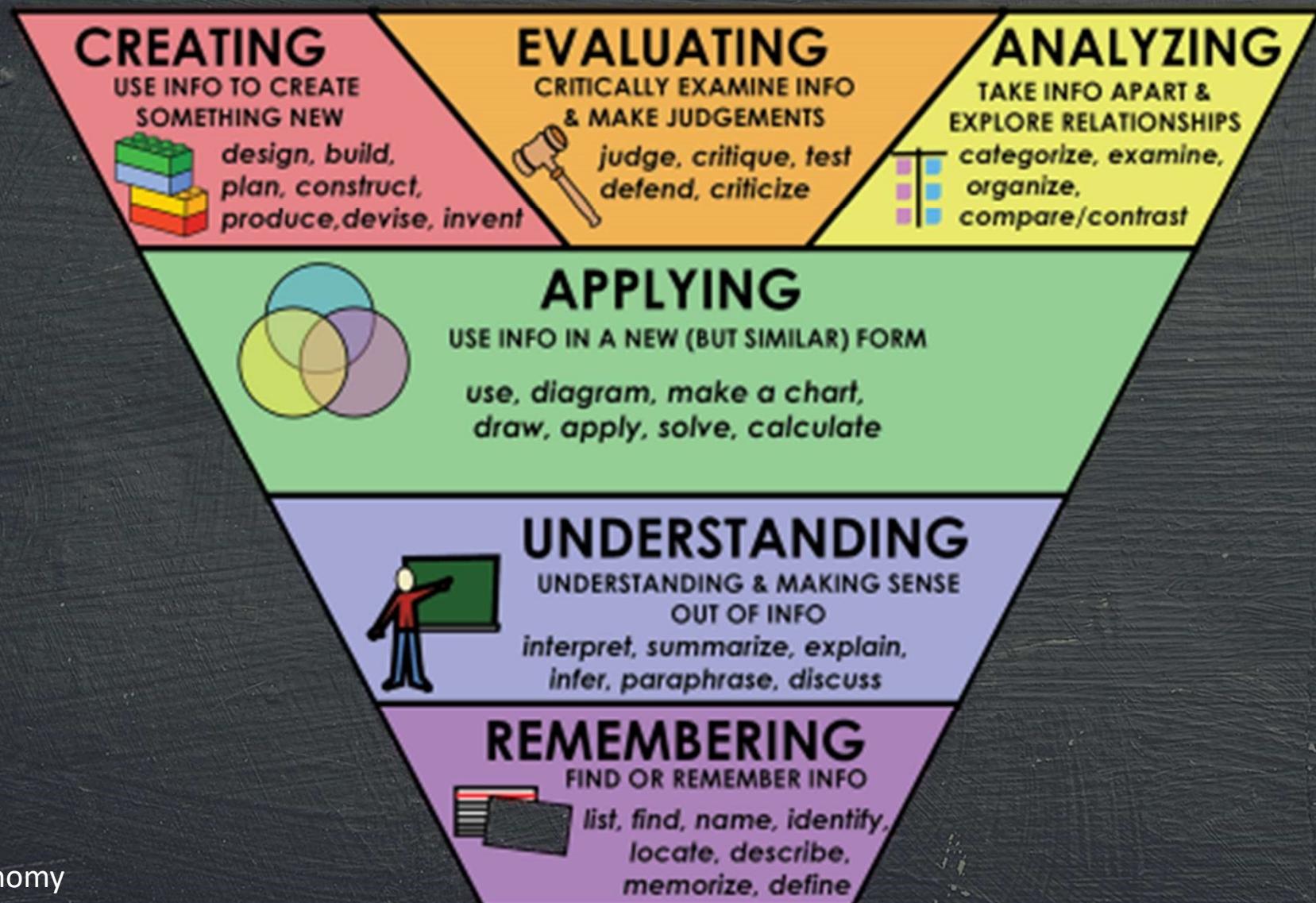
# Learning Through Tinkering

- What should we be learning?
- Apply what you learned... but how?
- Why do we stop before it's finished?
- When should we be tinkering?

Growing Emphasis on  
Learning in  
Personal Time

“Should your employer  
give you time to tinker?”

YES



# It is ALREADY a WIN/WIN Story

- Motivation
- Higher self-esteem
- Creativity
- ...
- Less turnover
- Grow company from within
- Better Productivity
- ...

Share what you  
learned!

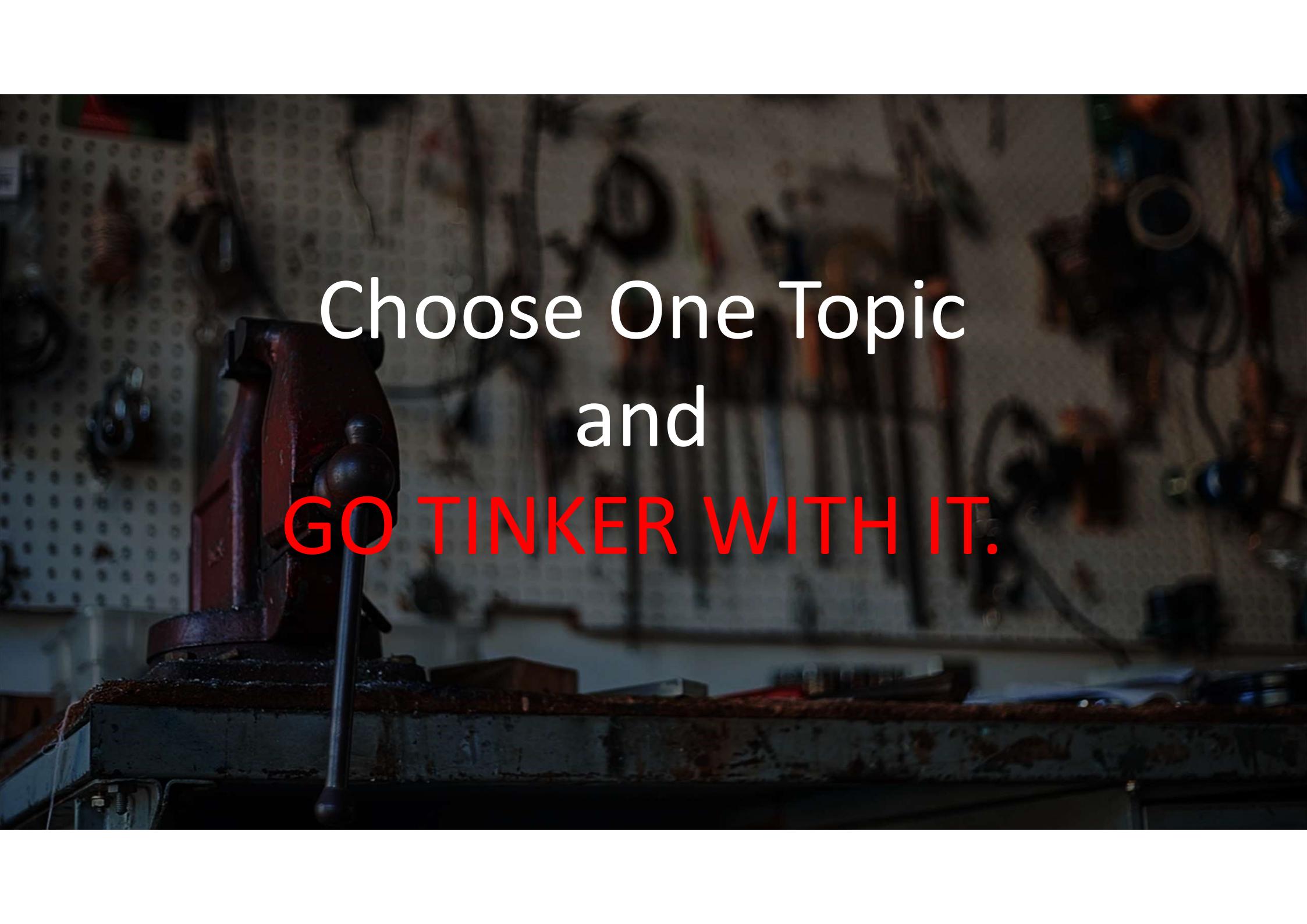
“Who is responsible to update and  
maintain your knowledge and skills?”

You Are



# Learning Through Tinkering

- What should we be learning?
- Apply what you learned... but how?
- Why do we stop before it's finished?
- When should we be tinkering?



Choose One Topic  
and  
**GO TINKER WITH IT.**



Bob Ross

1,897,404 subscribers

HOME

VIDEOS

PLAYLISTS

COMMUNITY

CHANNELS

## About this talk



Evil consequences

Squad



meetup  
The official JUG  
of Utrecht

# Thank You

Tom Cools  
@TCoolsIT  
[www.tomcools.be](http://www.tomcools.be)

Everything is available online <3  
Feedback much appreciated!