



**BERLIN
BUZZWORDS
2019** JUNE 16-18

**“Band-aids don’t fix bullet holes:”
repairing the broken promises of
ubiquitous machine learning**

William Benton • @willb • willb@redhat.com





HUGE

HUGE

HUGE

LOGO

LOGO

LOGO

**MEDIUM
LOGO**

**MEDIUM
LOGO**

**MEDIUM
LOGO**

**MEDIUM
LOGO**

**MEDIUM
LOGO**

tiny logo

tiny logo

tiny logo

tiny logo

tiny logo

tiny logo

tiny logo

tiny logo

tiny logo

tiny logo

tiny logo

tiny logo

tiny logo

tiny logo

'CAUSE BABY,
NOW WE'VE GOT
BAD BLOOD

HUGE

YOU KNOW IT USED
TO BE **MAD LOVE...**

NOW WE'VE GOT
PROBLEMS AND I
DON'T THINK WE
CAN **SOLVE THEM**

HUGE

HUGE

HUGE

LOGO

LOGO

LOGO

**MEDIUM
LOGO**

**MEDIUM
LOGO**

**MEDIUM
LOGO**

**MEDIUM
LOGO**

**MEDIUM
LOGO**

tiny logo

tiny logo

tiny logo

tiny logo

tiny logo

tiny logo

tiny logo

tiny logo

tiny logo

tiny logo

tiny logo

tiny logo

tiny logo

tiny logo

**HUGE
LOGO**



**HUGE
LOGO**

**HUGE
LOGO**

**MEDIUM
LOGO**

**MEDIUM
LOGO**

**MEDIUM
LOGO**

**MEDIUM
LOGO**

**MEDIUM
LOGO**

tiny logo

tiny logo

tiny logo

tiny logo

tiny logo

tiny logo

tiny logo

tiny logo

tiny logo

tiny logo

tiny logo

tiny logo

tiny logo

tiny logo

**HUGE
LOGO**

**HUGE
LOGO**

**HUGE
LOGO**

**MEDIUM
LOGO**

**MEDIUM
LOGO**

**MEDIUM
LOGO**

**MEDIUM
LOGO**

**MEDIUM
LOGO**

tiny logo

tiny logo

tiny logo

tiny logo

tiny logo

tiny logo

tiny logo

tiny logo

tiny logo

tiny logo

tiny logo

tiny logo

tiny logo

tiny logo

**HUGE
LOGO**

**HUGE
LOGO**

**HUGE
LOGO**

**MEDIUM
LOGO**

**MEDIUM
LOGO**

**MEDIUM
LOGO**

**MEDIUM
LOGO**

**MEDIUM
LOGO**

tiny logo

tiny logo

tiny logo

tiny logo

tiny logo

tiny logo

tiny logo

tiny logo

tiny logo

tiny logo

tiny logo

tiny logo

tiny logo

tiny logo

HUGE
LOGO

HUGE
LOGO

HUGE
LOGO

MEDIUM
LOGO

MEDIUM
LOGO

MEDIUM
LOGO

MEDIUM
LOGO

MEDIUM
LOGO

tiny logo

tiny logo

tiny logo

tiny logo

tiny logo

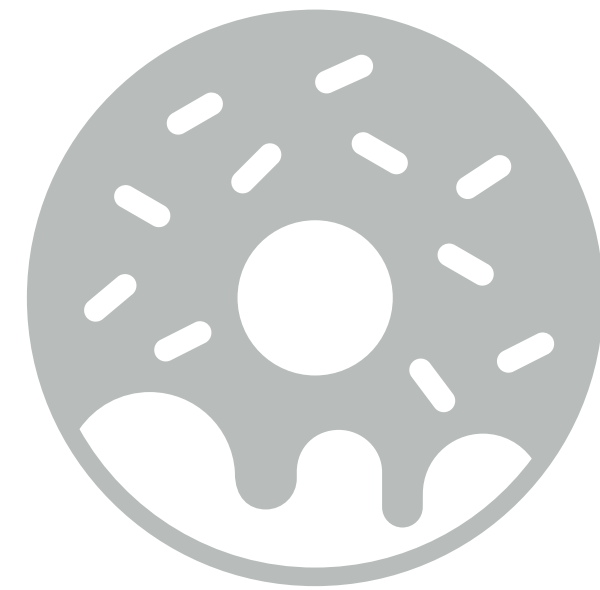
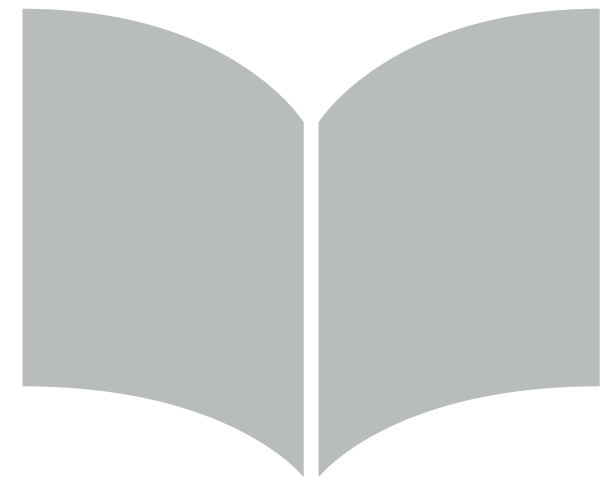
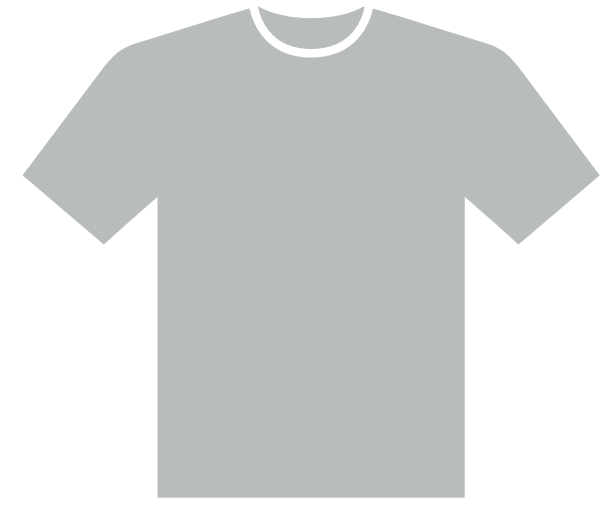
tiny logo

tiny logo

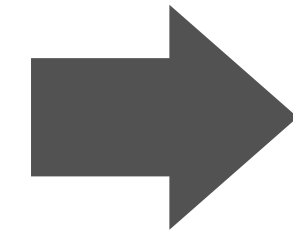
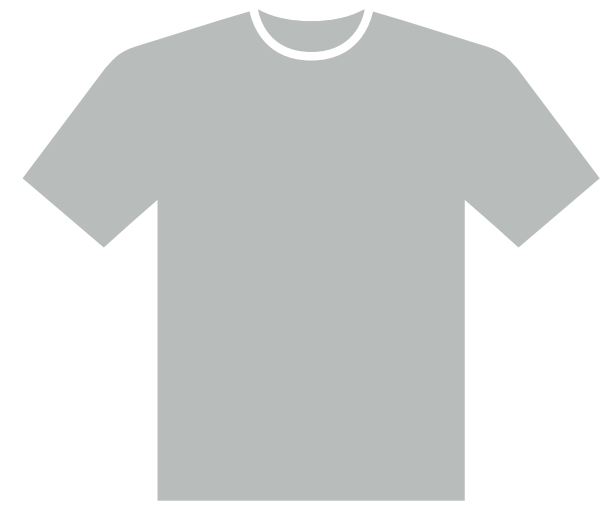
tiny logo

tiny logo

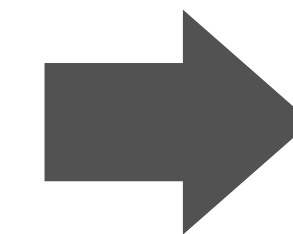
**BAND-AIDS
DON'T FIX
BULLET HOLES**



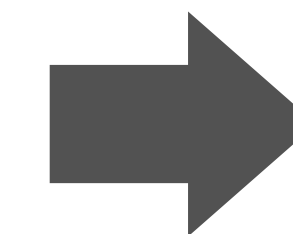
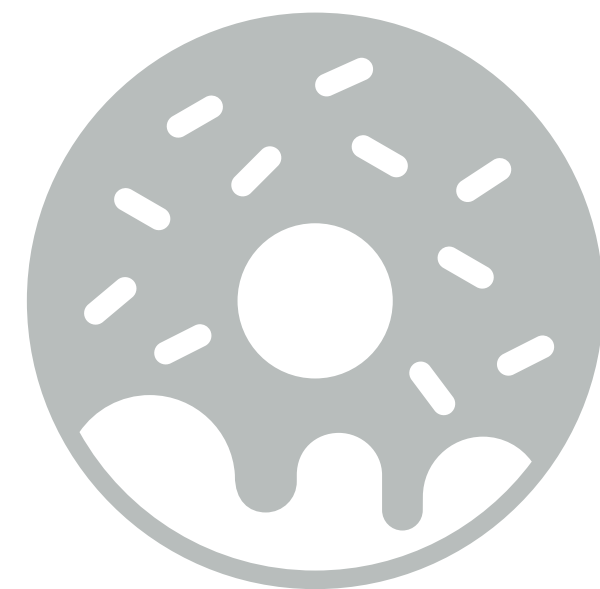
@willb #bbuzz



[0.07 0.06 ... 0.50 0.34]



[0.18 0.25 ... 0.45 0.99]



[0.92 0.29 ... 0.19 0.61]



```
apparel_model = load_embedding("apparel.hdf5")
book_model = load_embedding("books.hdf5")
pastry_model = load_embedding("pastries.hdf5")

def serve(clothing, book):
    av = apparel_model.score(clothing)
    bv = book_model.score(book)
    return(pastry_model.find_closest(av + bv))
```

```
apparel_model = load_embedding("apparel.hdf5")
book_model = load_embedding("book.hdf5")
pastry_model = load_embedding("pastry.hdf5")

def find_closest_pastry(av, bv):
    av = apparel_model[av]
    bv = book_model[bv]
    return pastry_model[pastry_model.find_closest(av + bv)]
```

**BAND-AIDS
DON'T FIX
BULLET HOLES**

Forecast

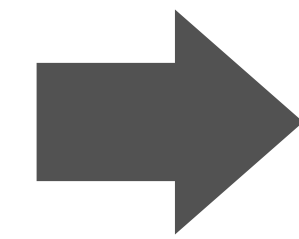
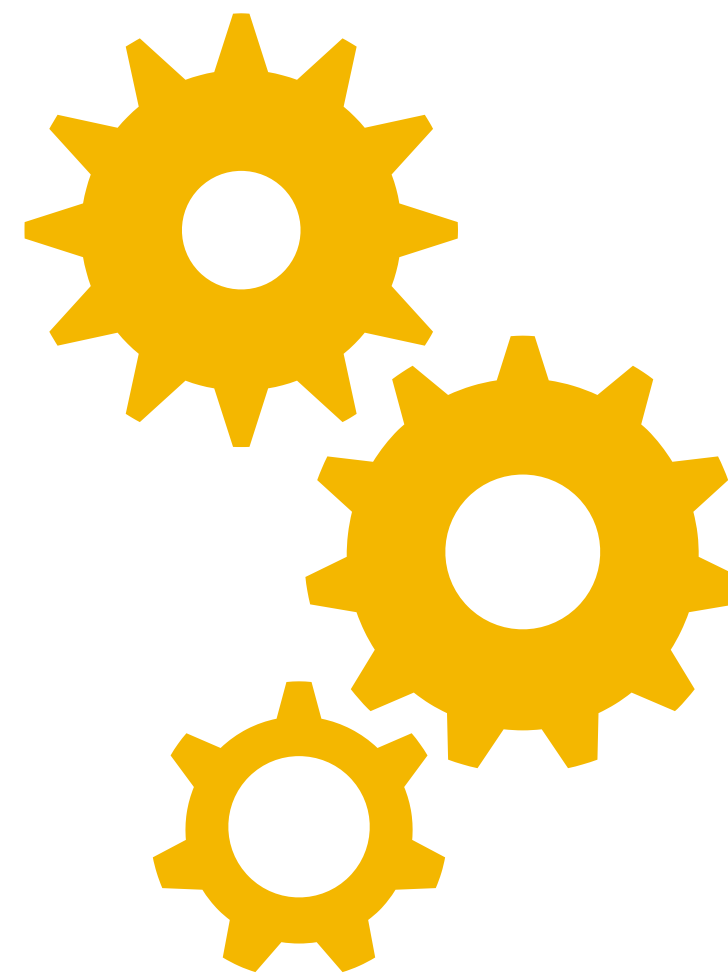
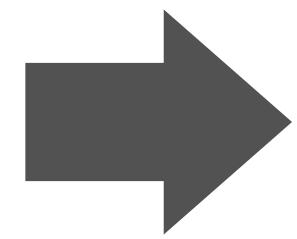
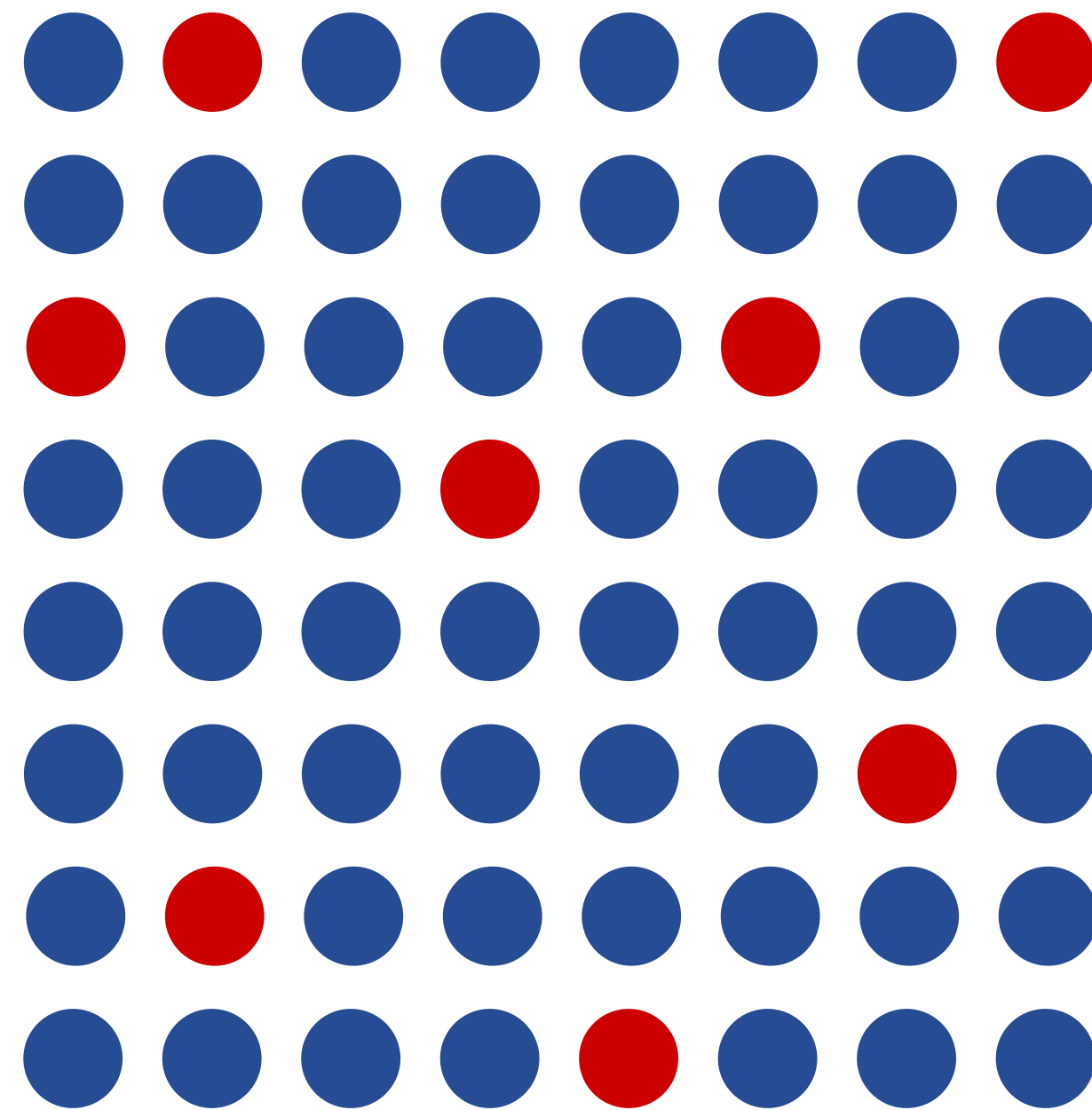
What's wrong with machine learning?

The wrong tools

The wrong incentives

The wrong problems

The way forward



**codifying problem
and metrics**

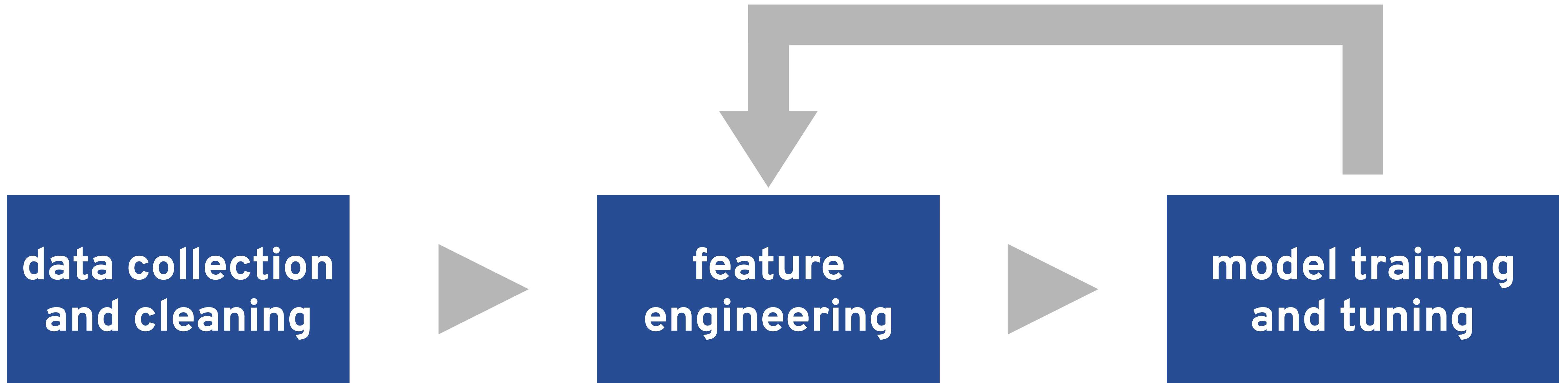


**data collection
and cleaning**

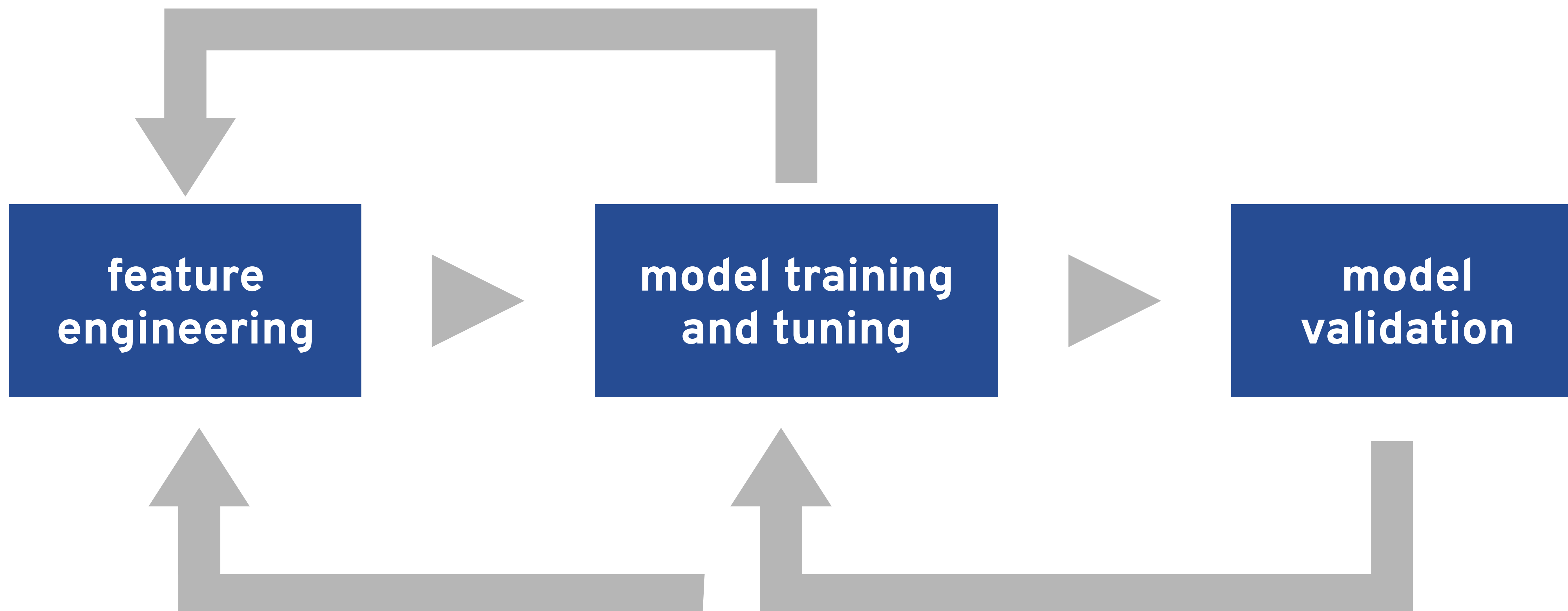
**data collection
and cleaning**

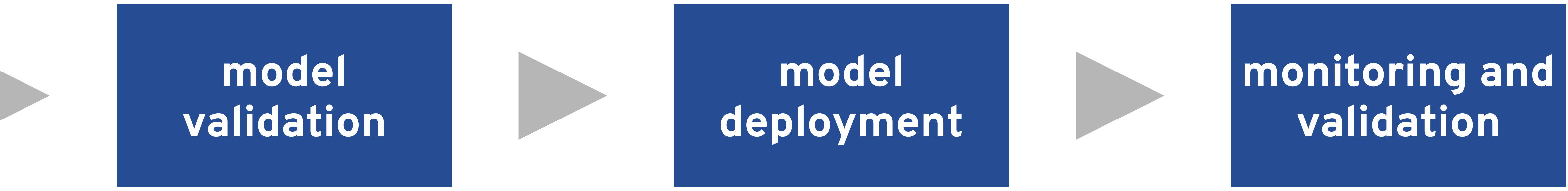
**feature
engineering**

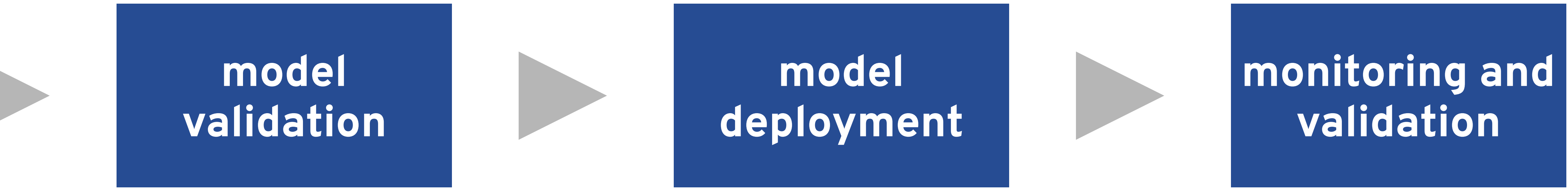
**model training
and tuning**



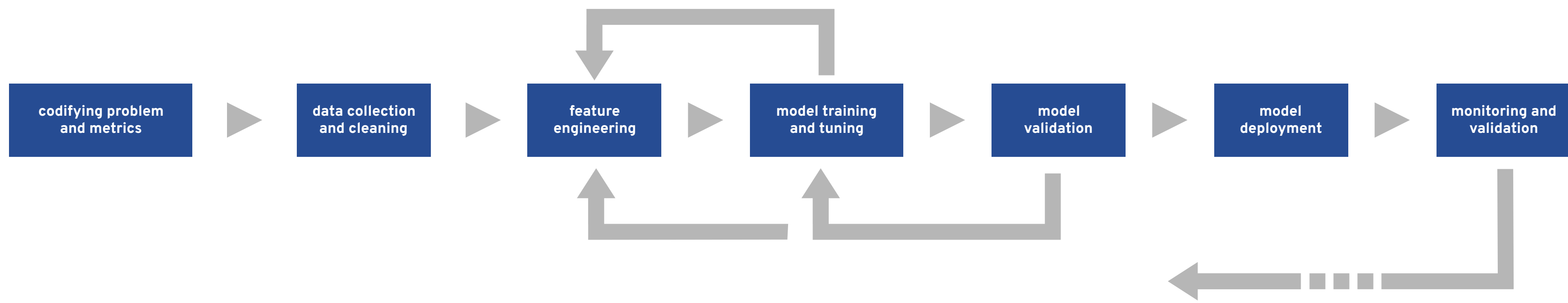




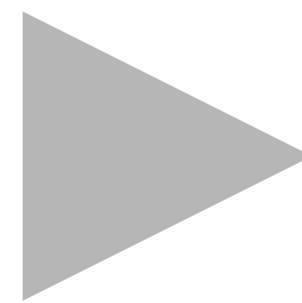




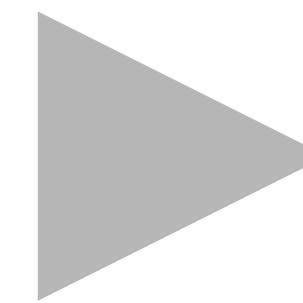
@willb #bbuzz



**data collection
and cleaning**

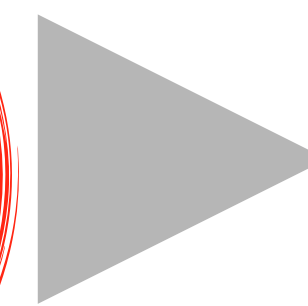
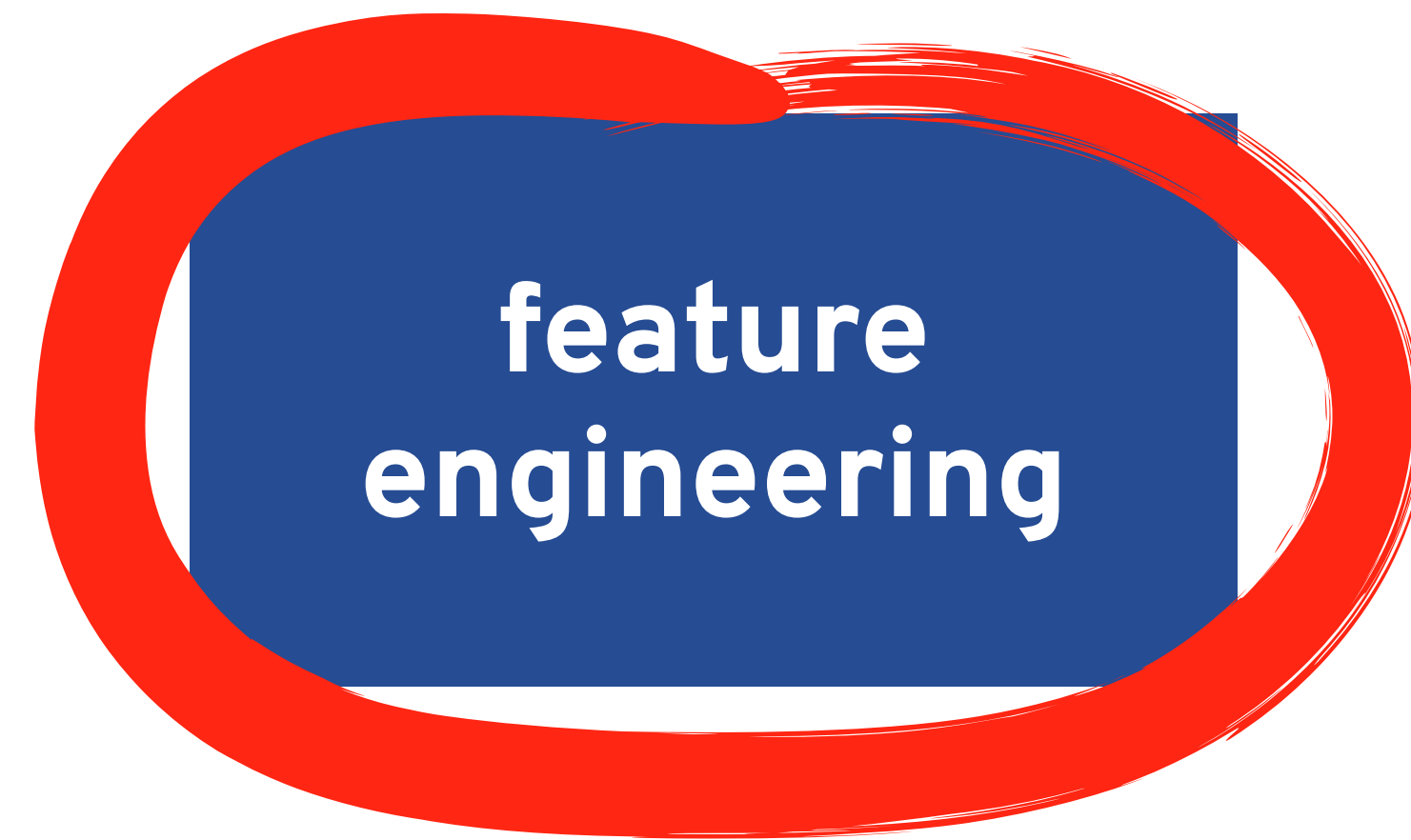
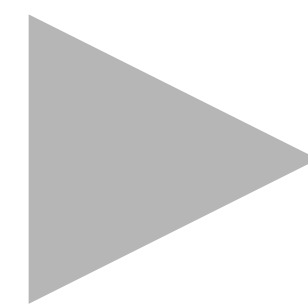


**feature
engineering**



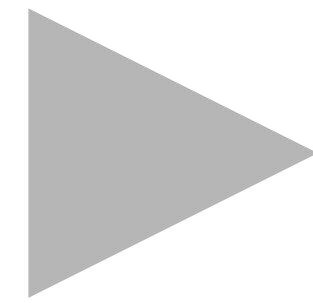
**model training
and tuning**

**data collection
and cleaning**

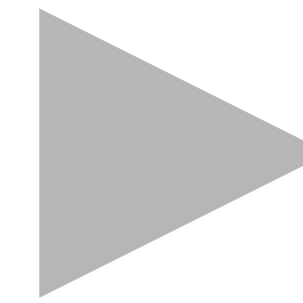


**model training
and tuning**

**feature
engineering**



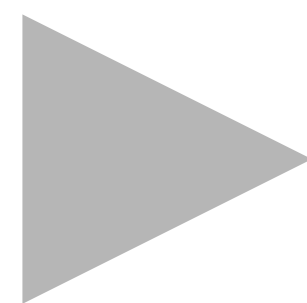
**model training
and tuning**



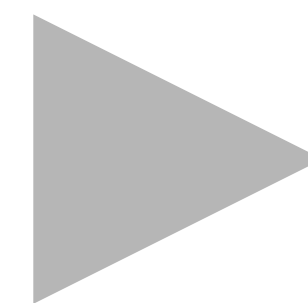
**model
validation**



**feature
engineering**

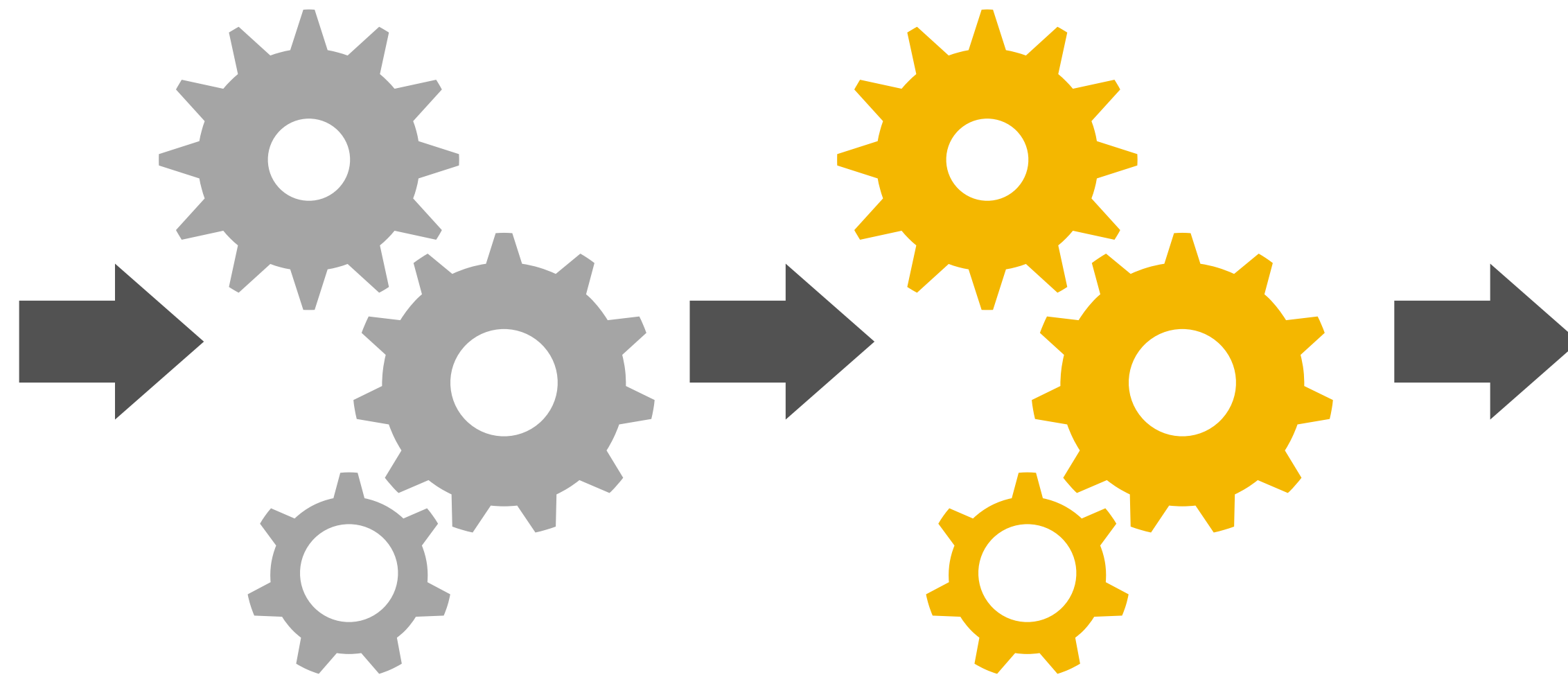
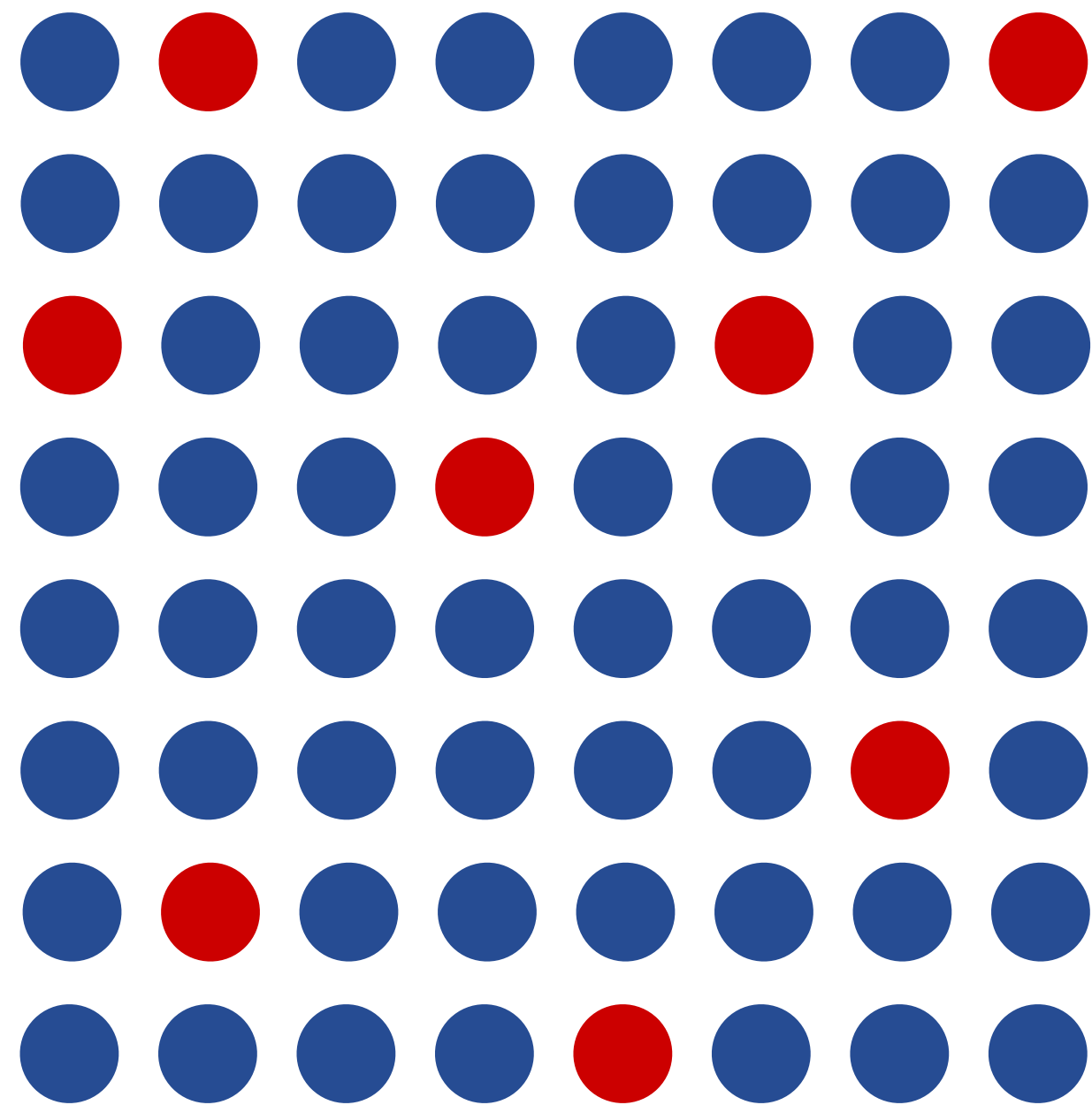


**model training
and tuning**



**model
validation**

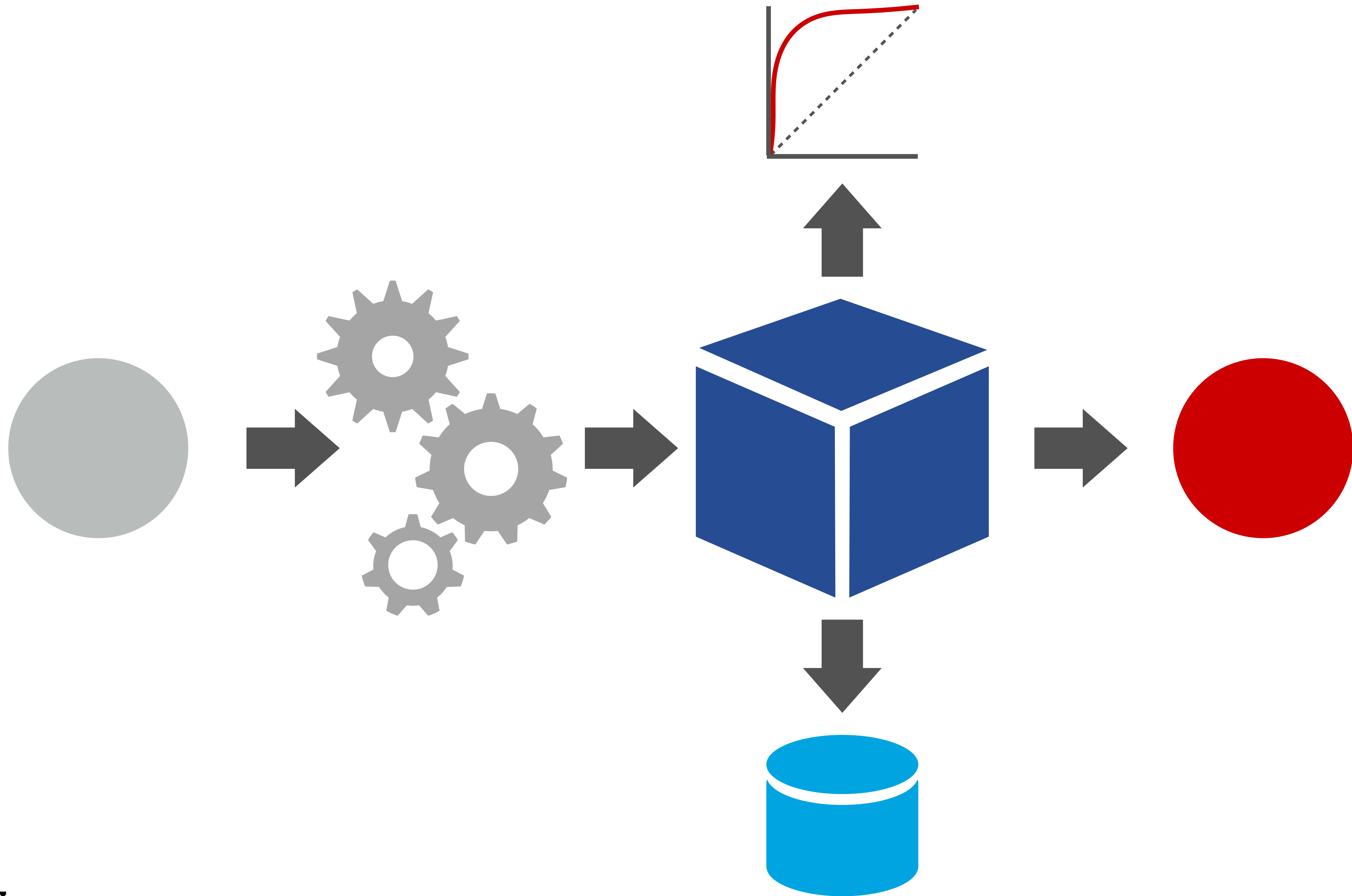




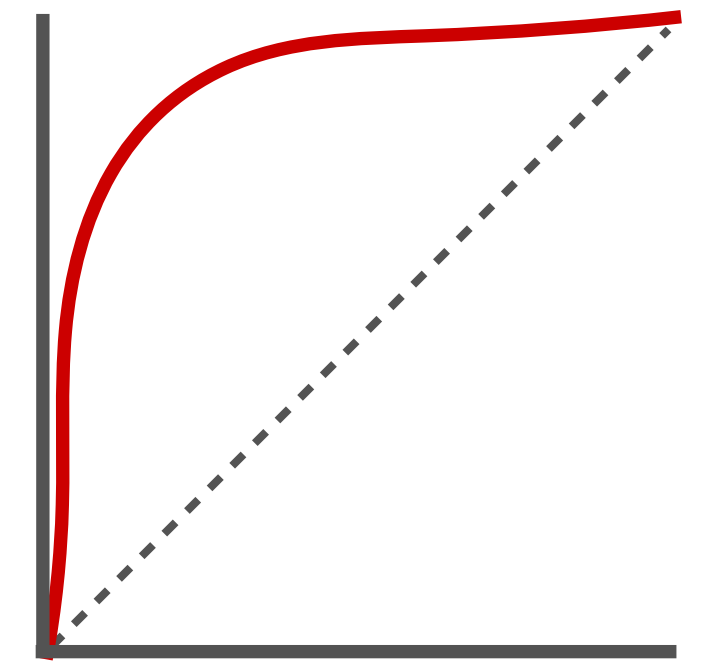
**feature
extraction**

**model
training**

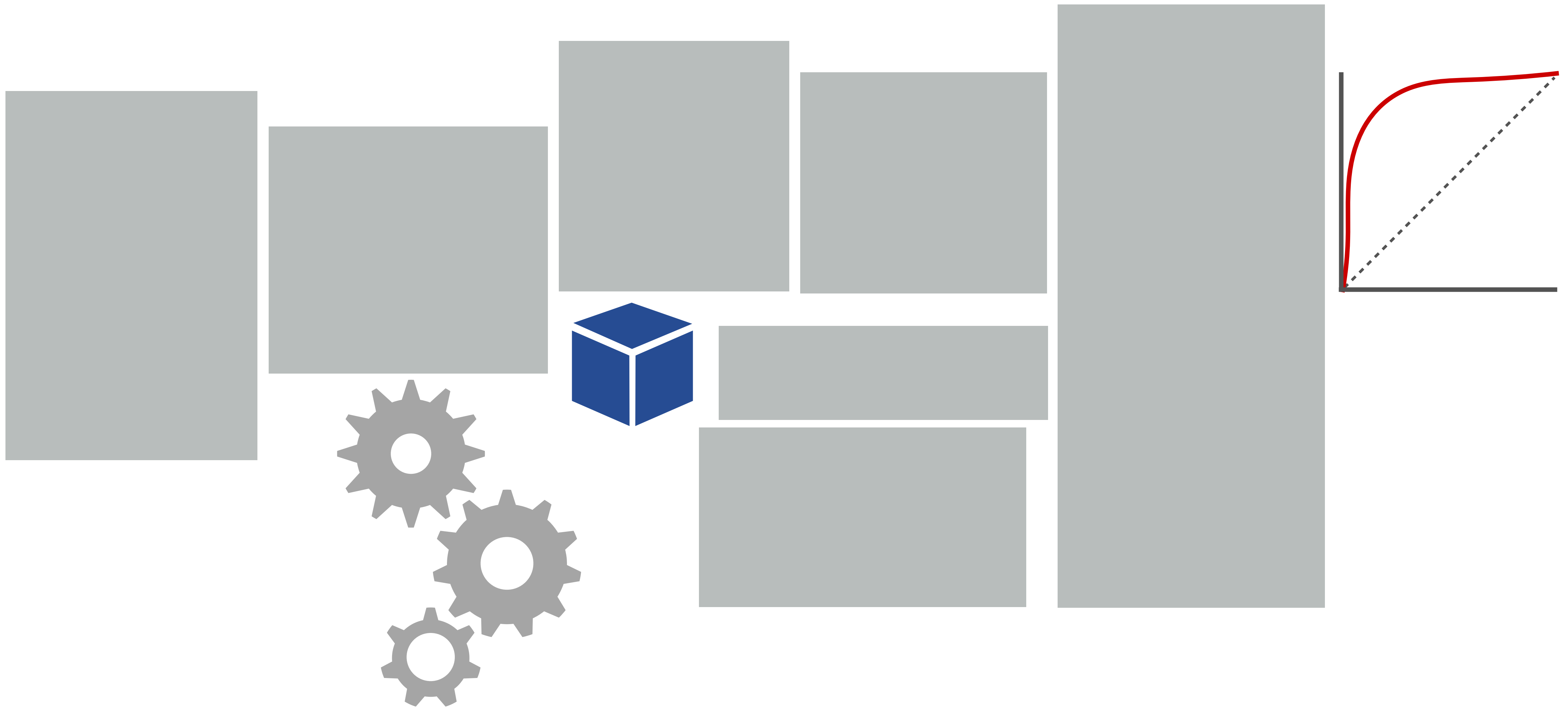




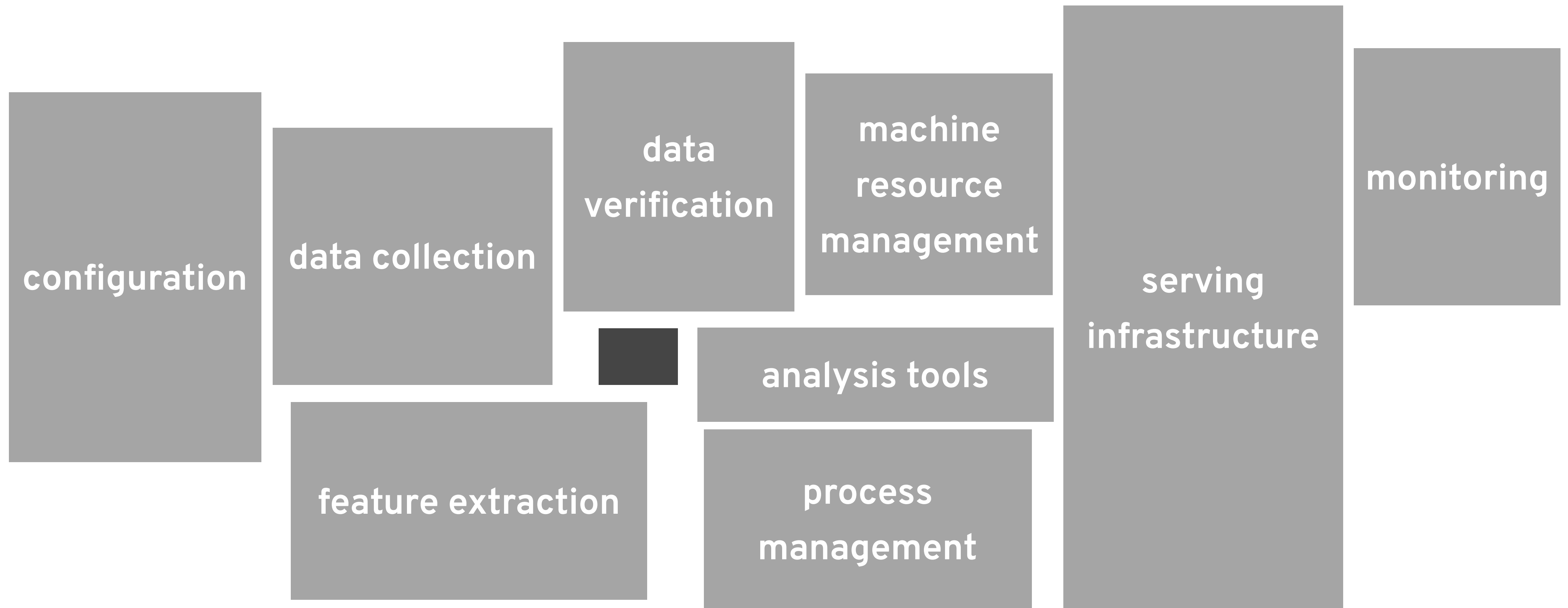
@willb #bbuzz



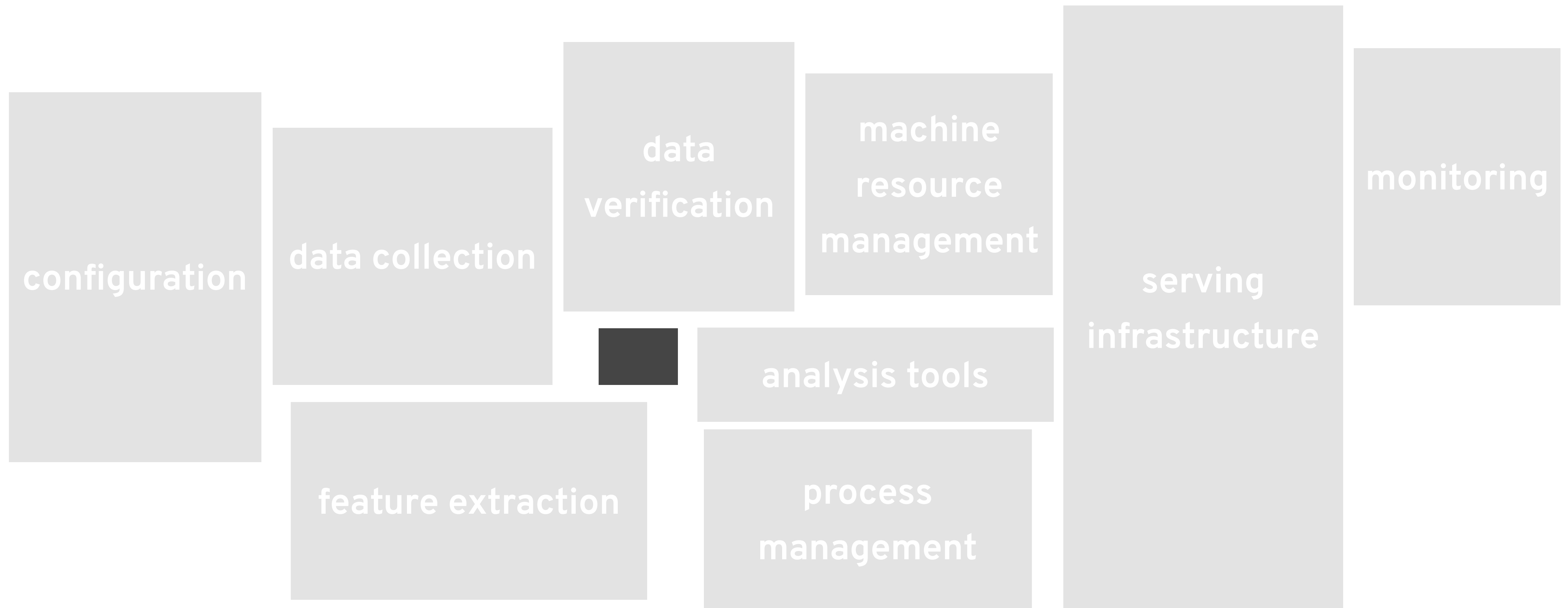
@willb #bbuzz



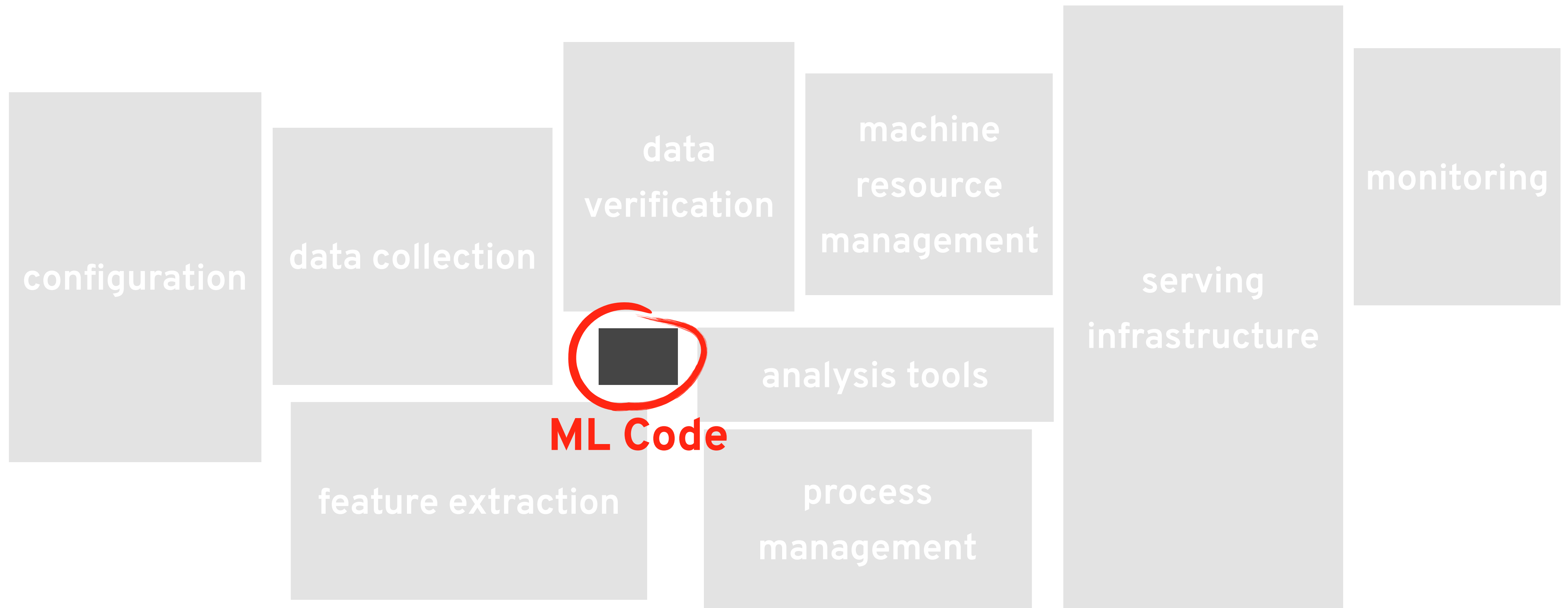
@willb #bbuzz



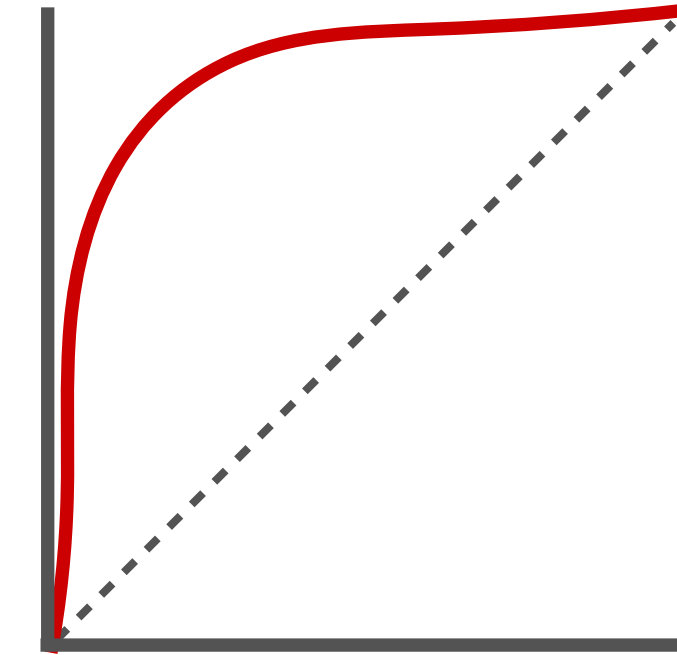
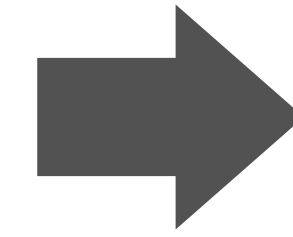
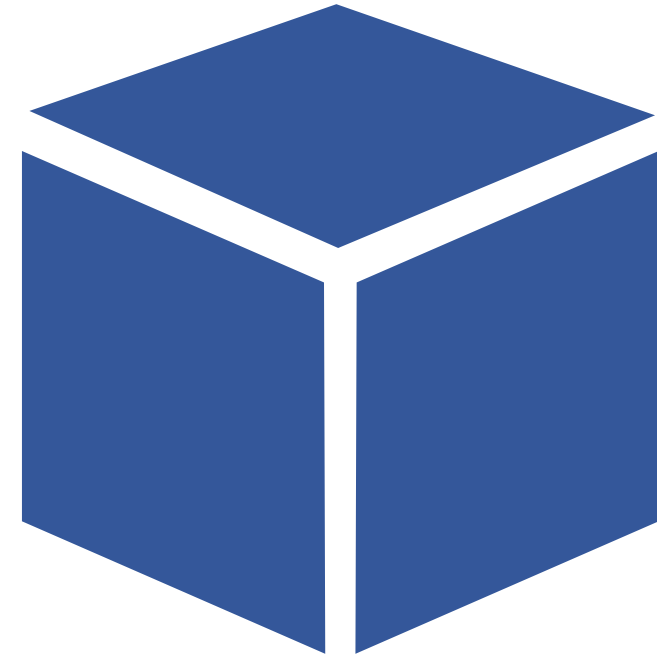
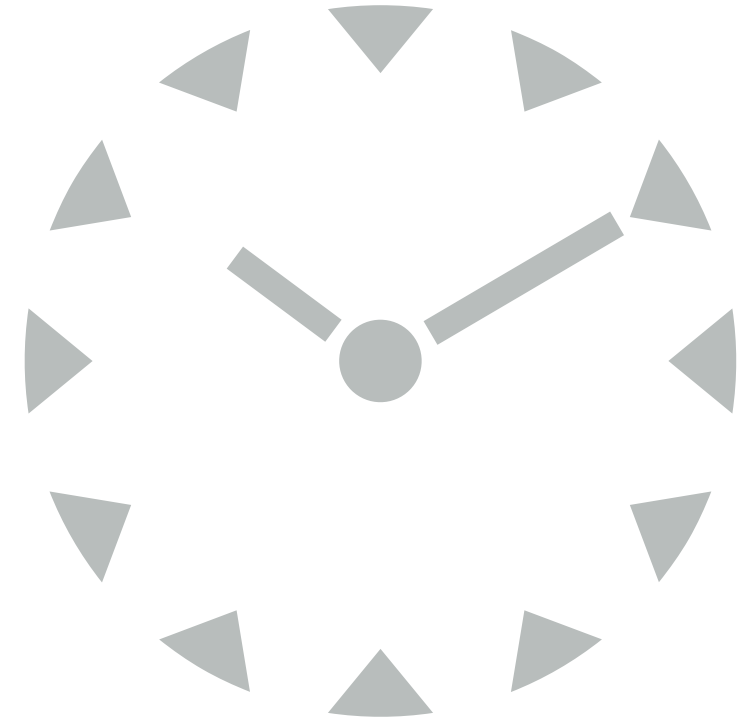
(Adapted from Sculley et al., “Hidden Technical Debt in Machine Learning Systems.” NIPS 2015)

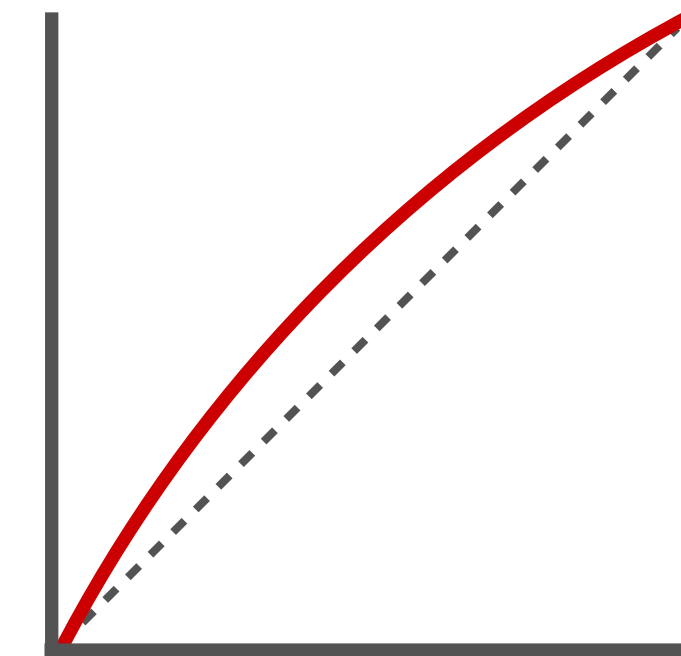
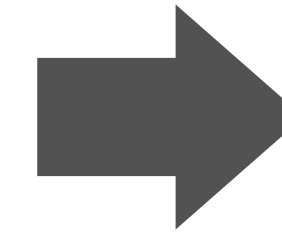
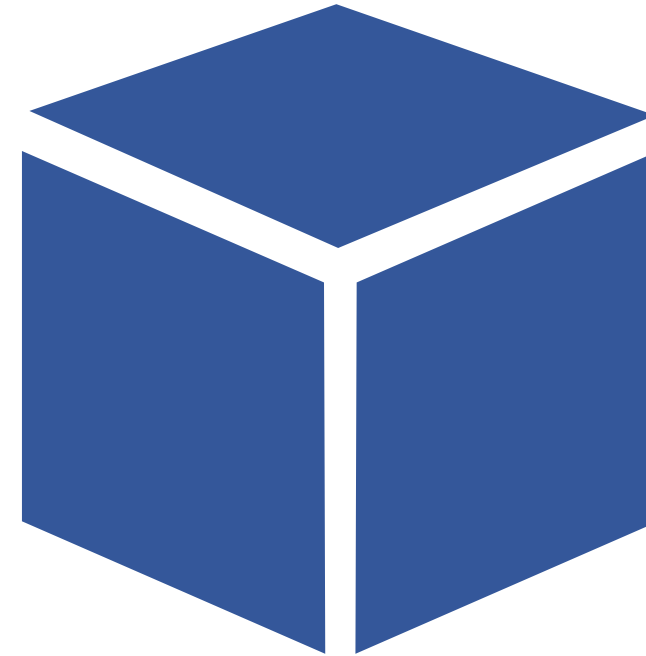
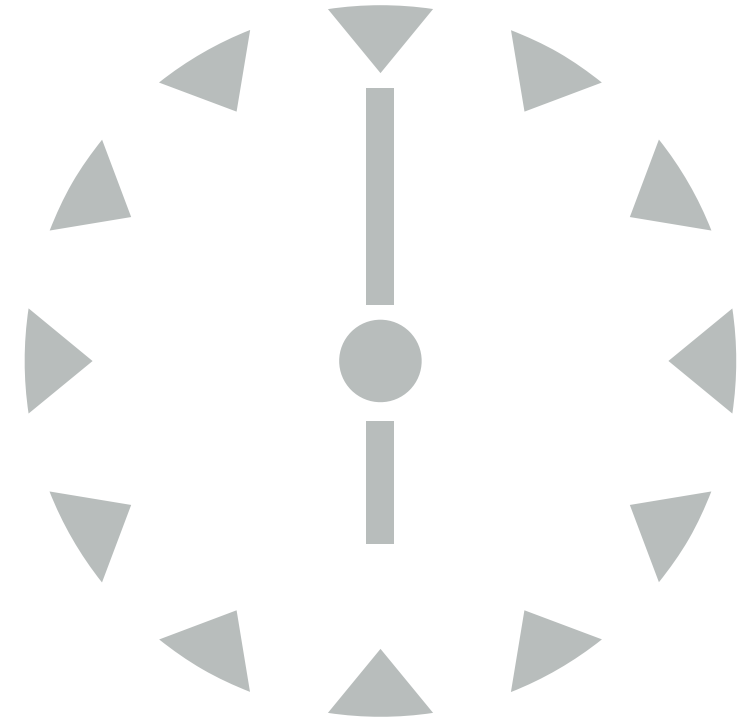


(Adapted from Sculley et al., "Hidden Technical Debt in Machine Learning Systems." NIPS 2015)



(Adapted from Sculley et al., "Hidden Technical Debt in Machine Learning Systems." NIPS 2015)







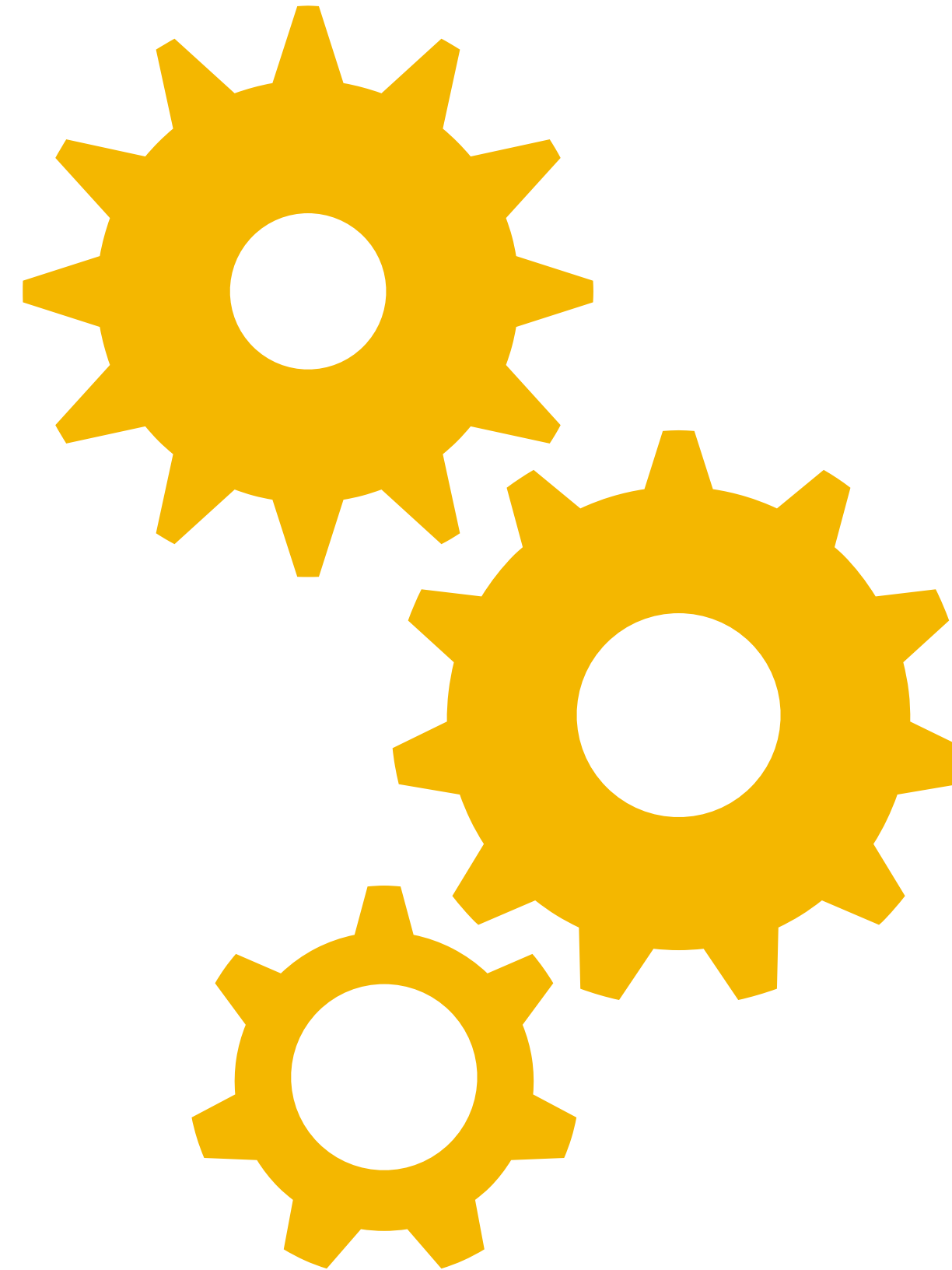


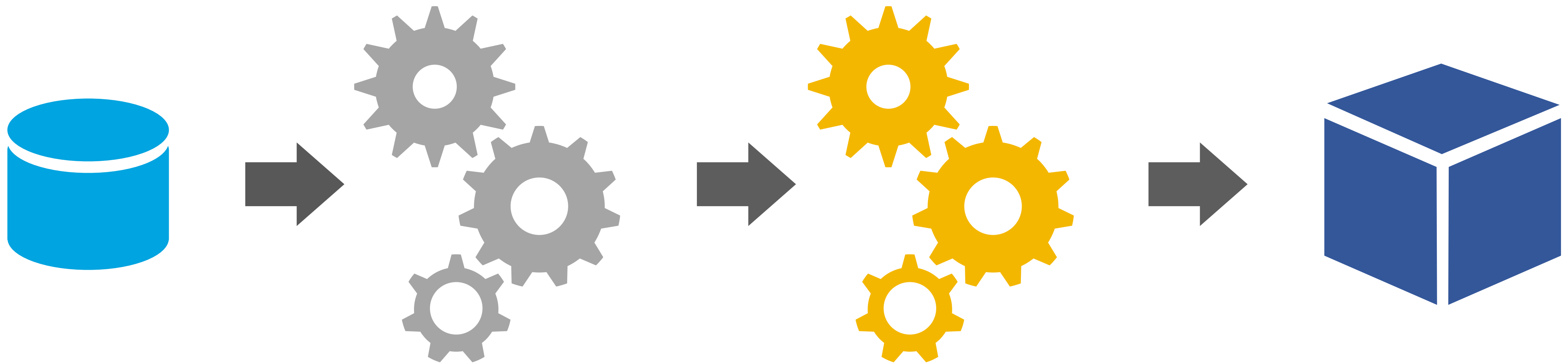




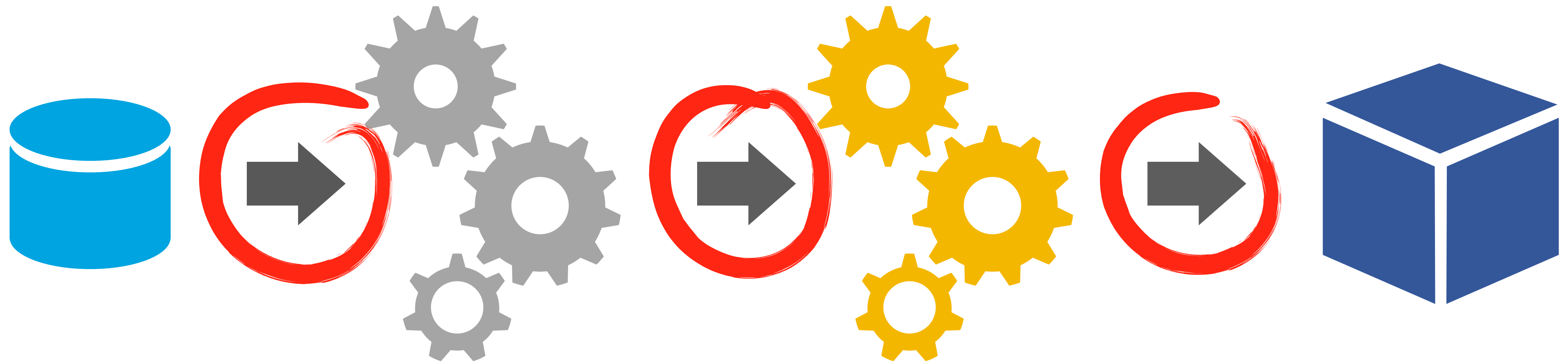
$C = 1000$

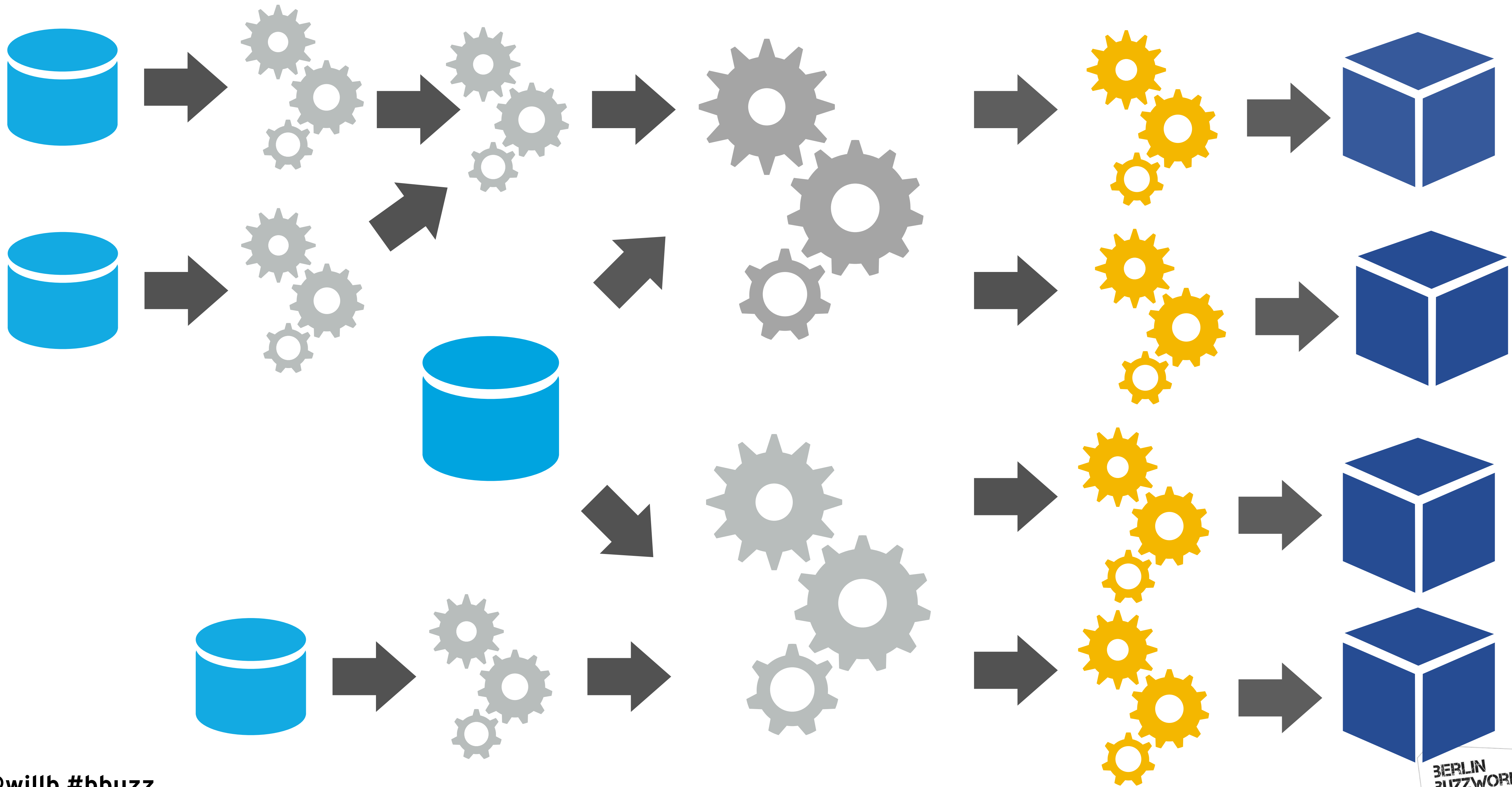
$\gamma = 0.1$





@willb #bbuzz



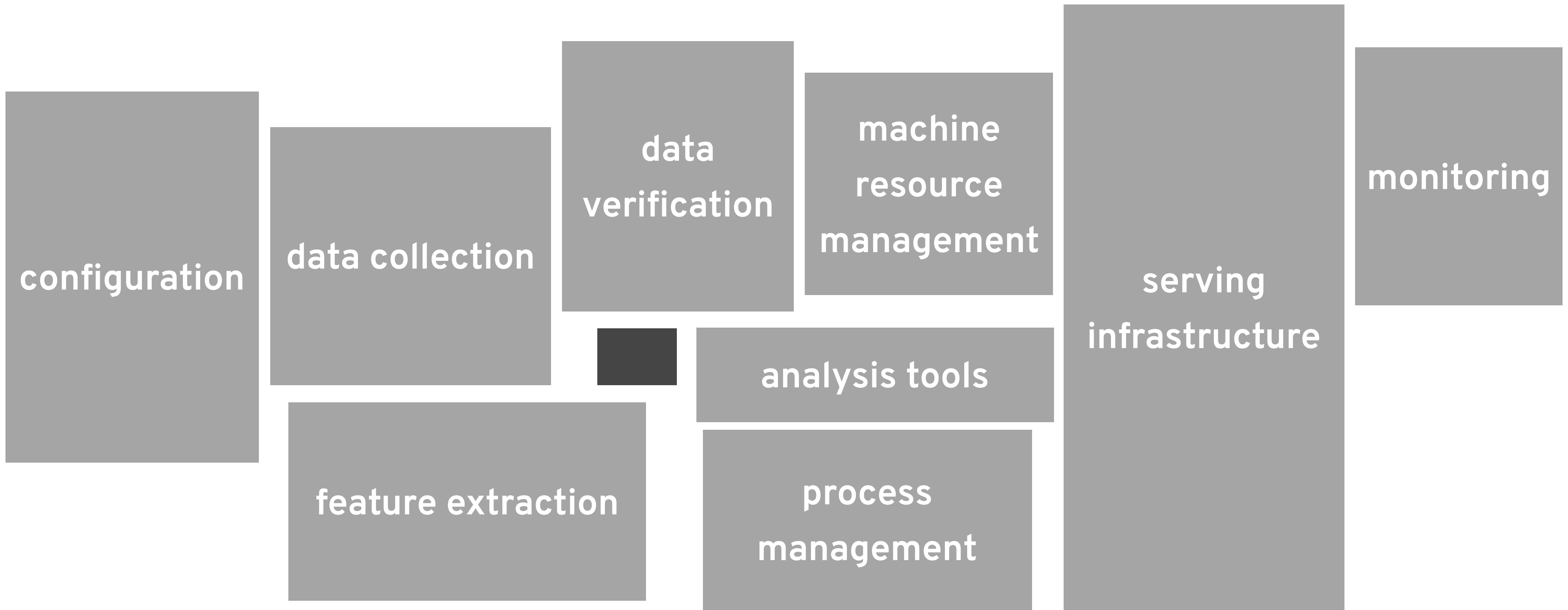


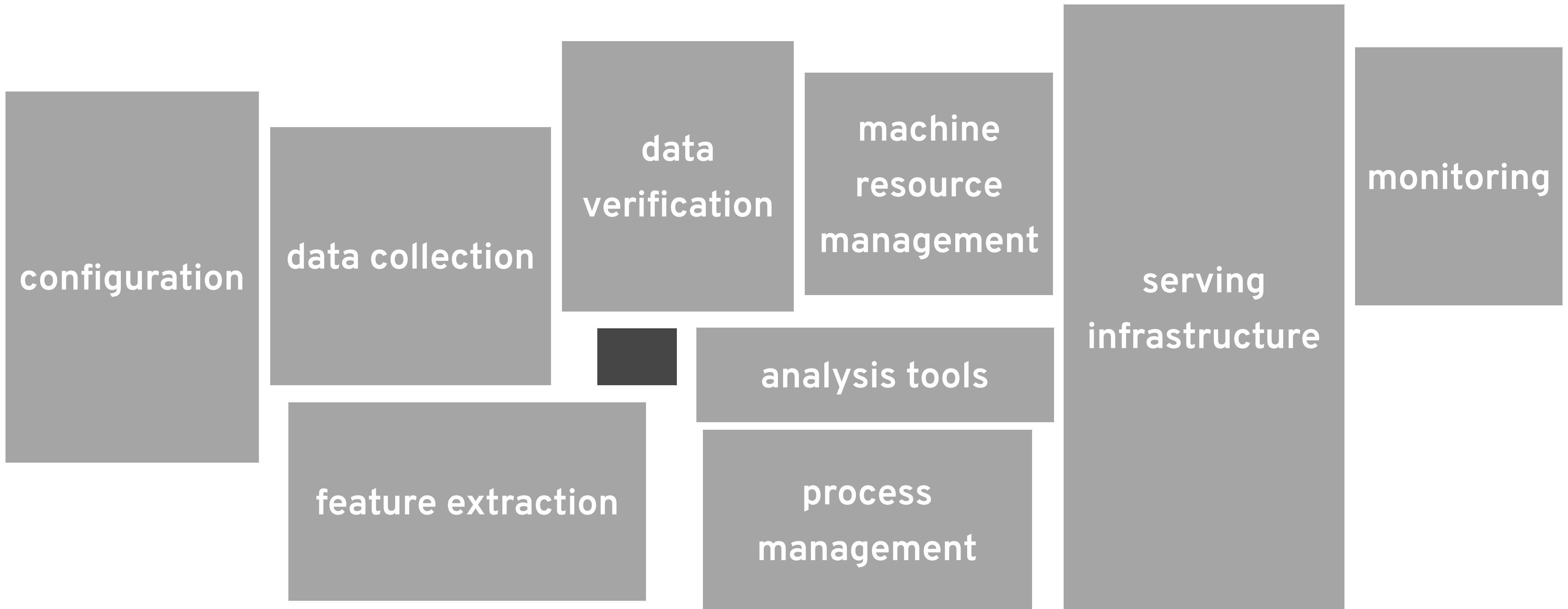
@willb #bbuzz

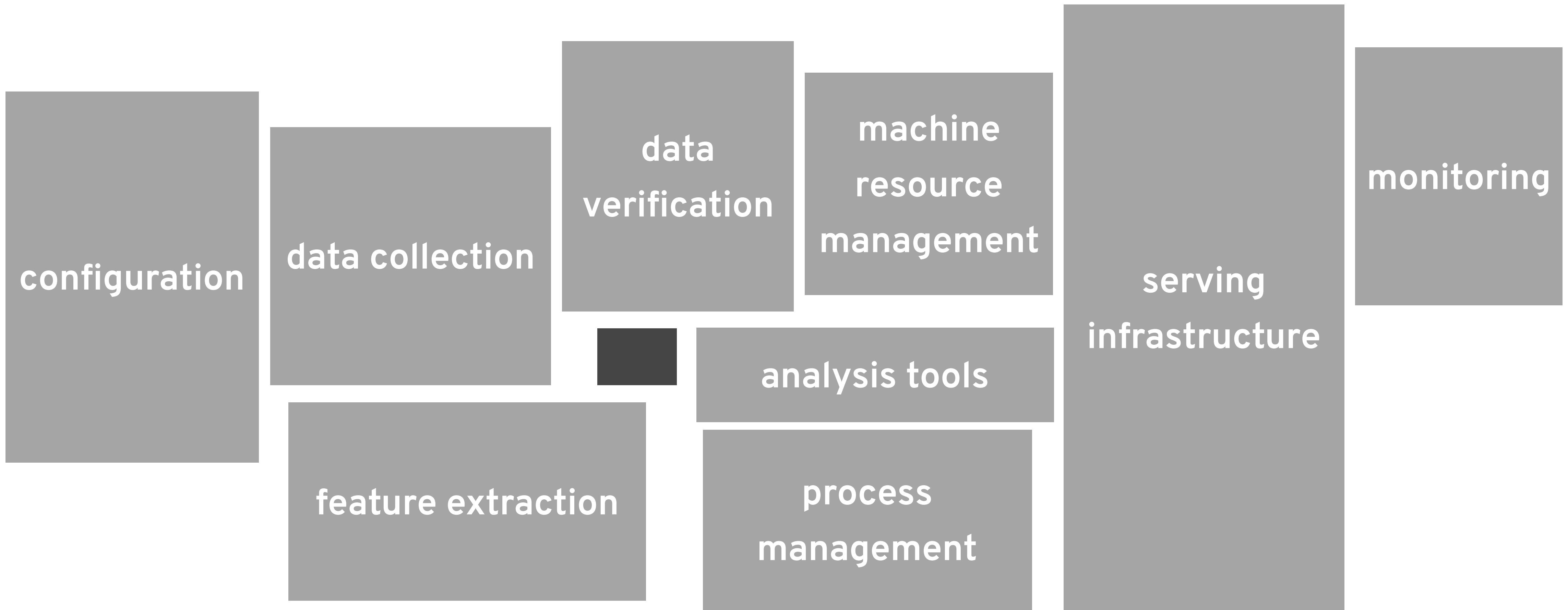
BERLIN
BUZZWORDS
2019 JUNE 16-18



Sculley et al., “Hidden Technical Debt in Machine Learning Systems.” *NIPS 2015*







Forecast

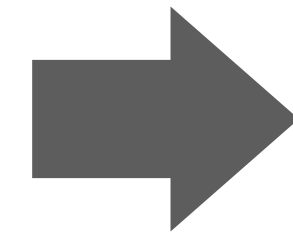
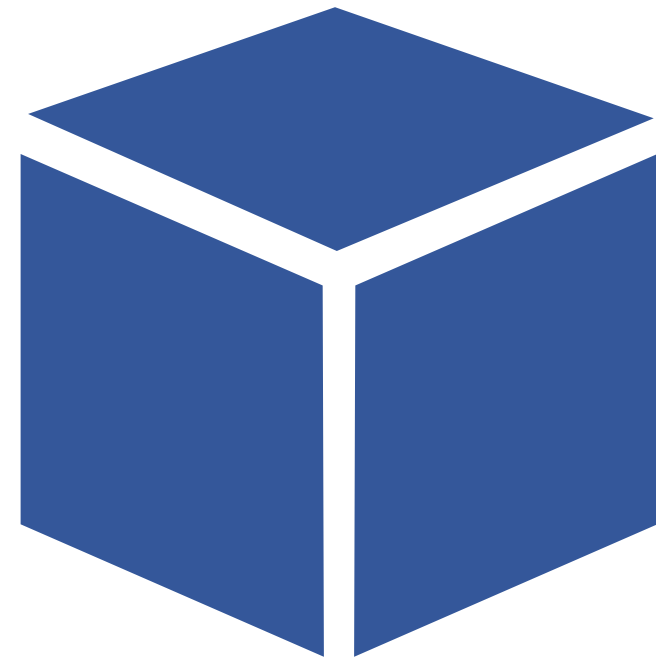
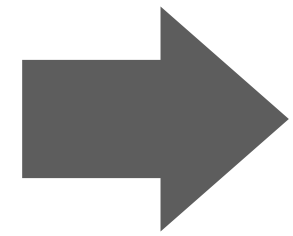
What's wrong with machine learning?

The wrong tools

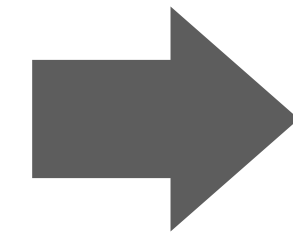
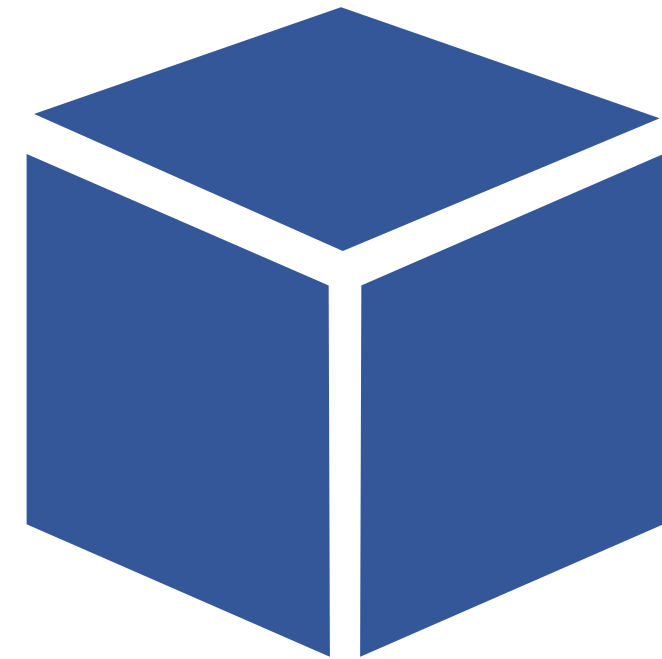
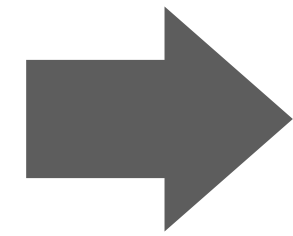
The wrong incentives

The wrong problems

The way forward



“face with heart
eyes emoji”



“face with heart
eyes emoji”



**codifying problem
and metrics**



**model training
and tuning**



**model
validation**

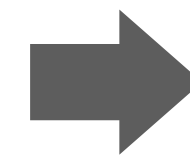
**codifying problem
and metrics**



**model training
and tuning**



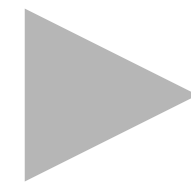
**model
validation**



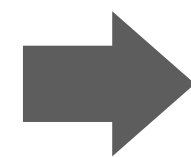
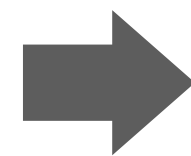
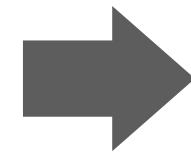
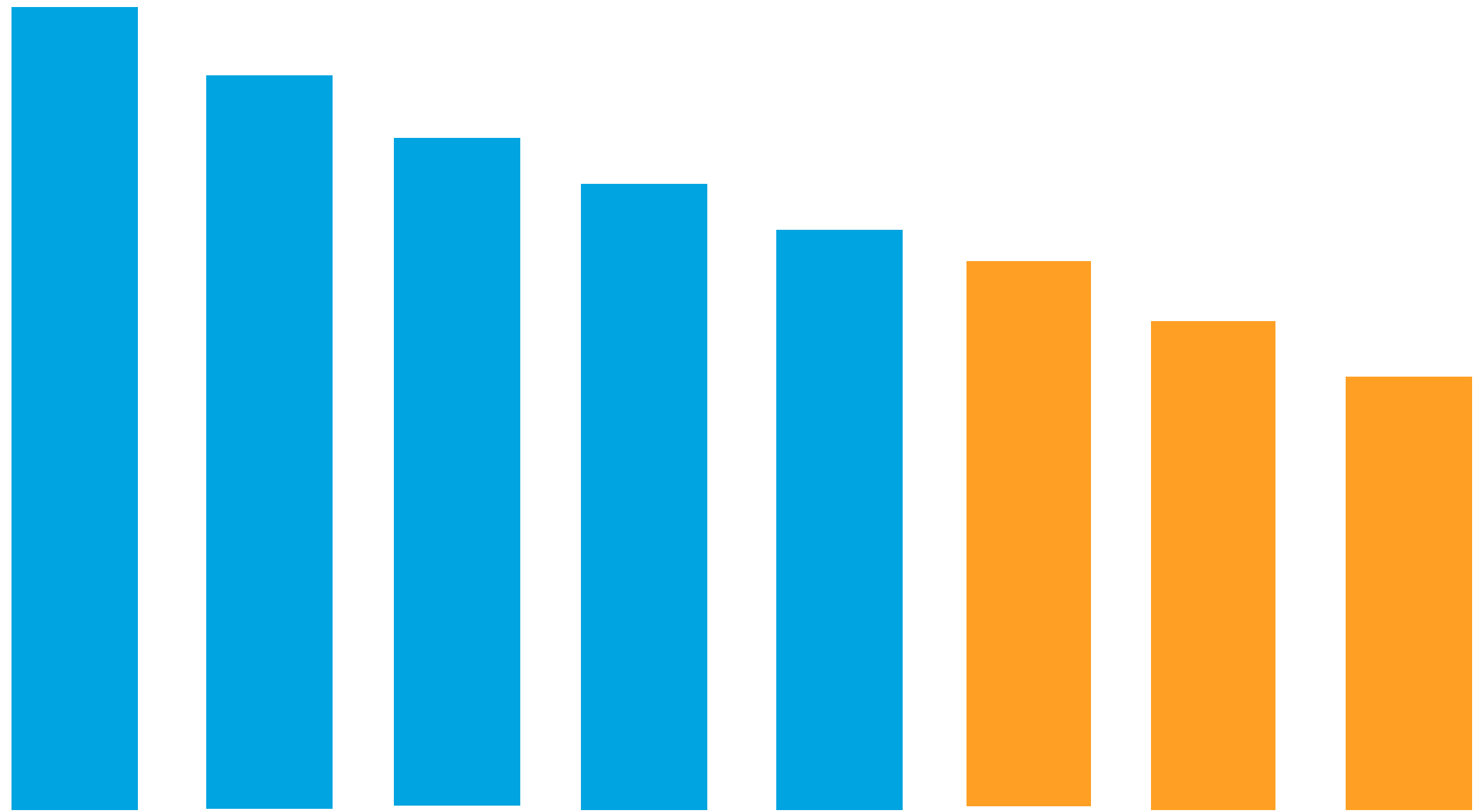
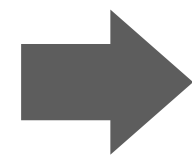
**codifying problem
and metrics**

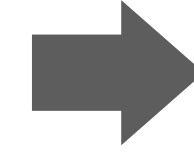
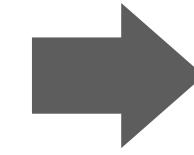
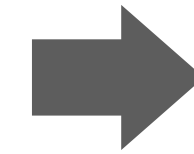
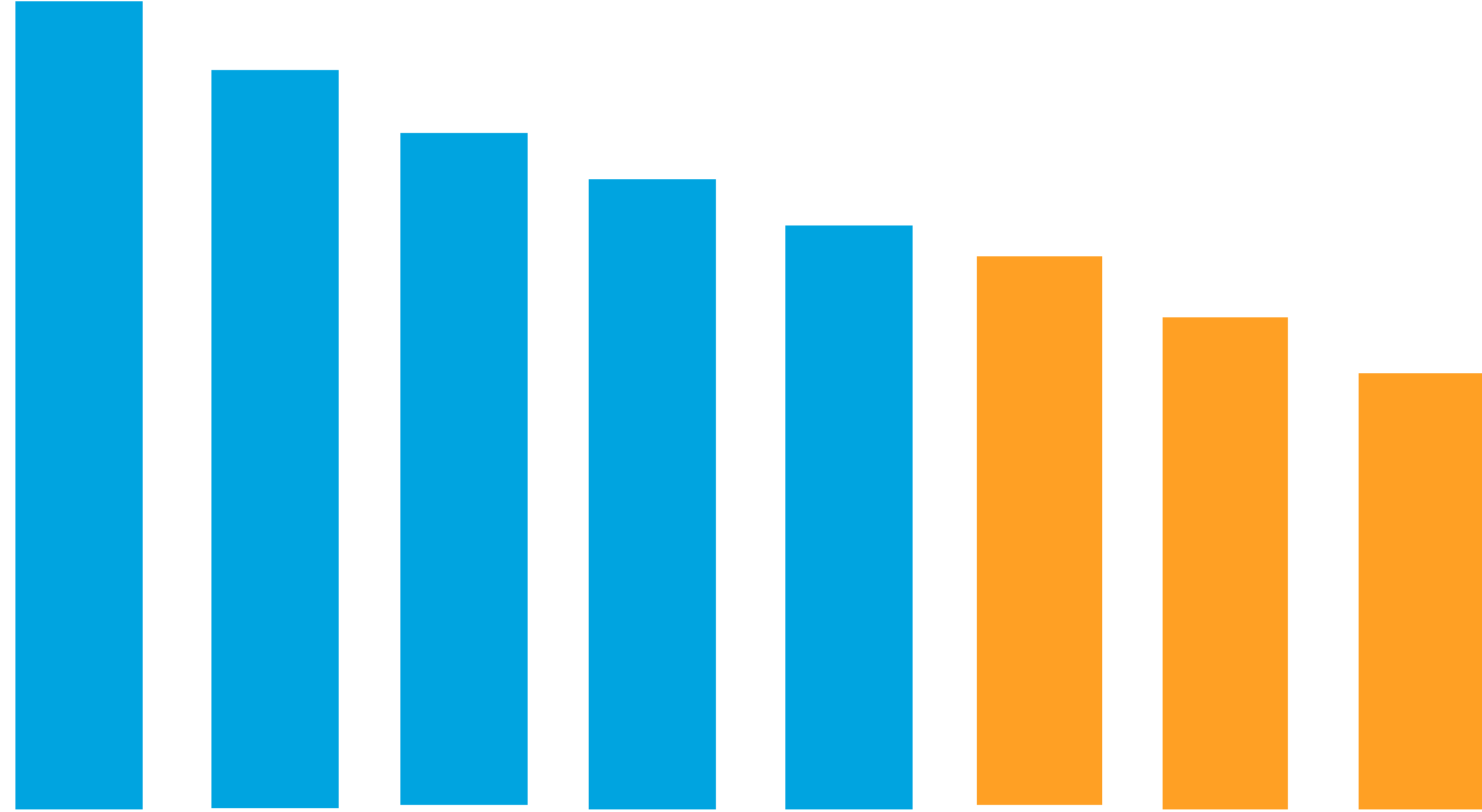
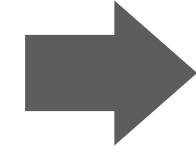


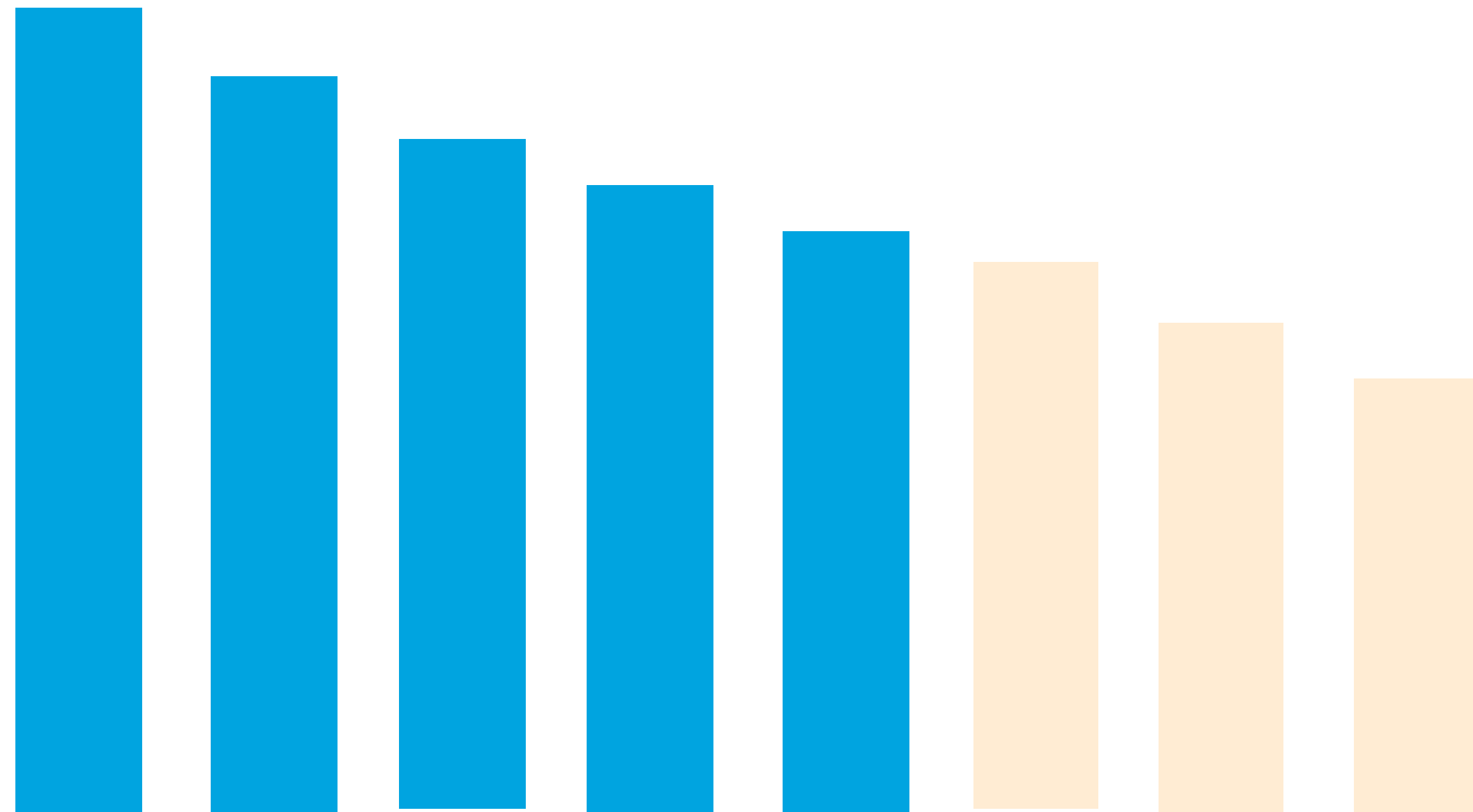
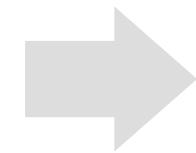
**model training
and tuning**



**model
validation**







$$f(x) = Y$$

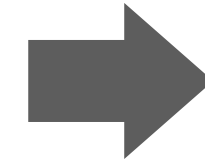
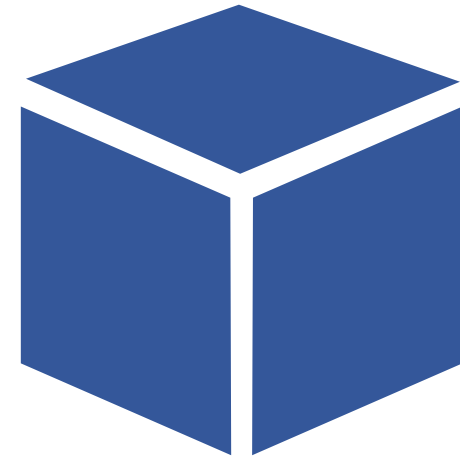
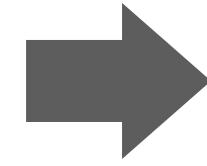


Szegedy et al., “Intriguing properties of neural networks.” In *ICLR '14*.

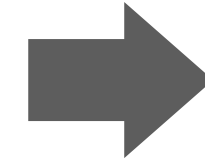
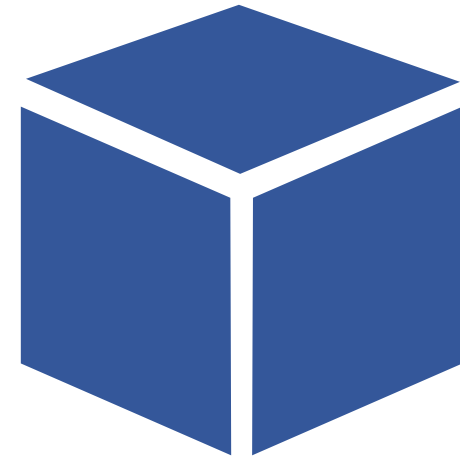
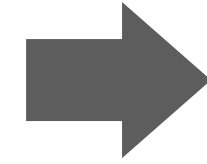
$$\frac{f(x) = Y}{f(x + \epsilon) = Y}$$



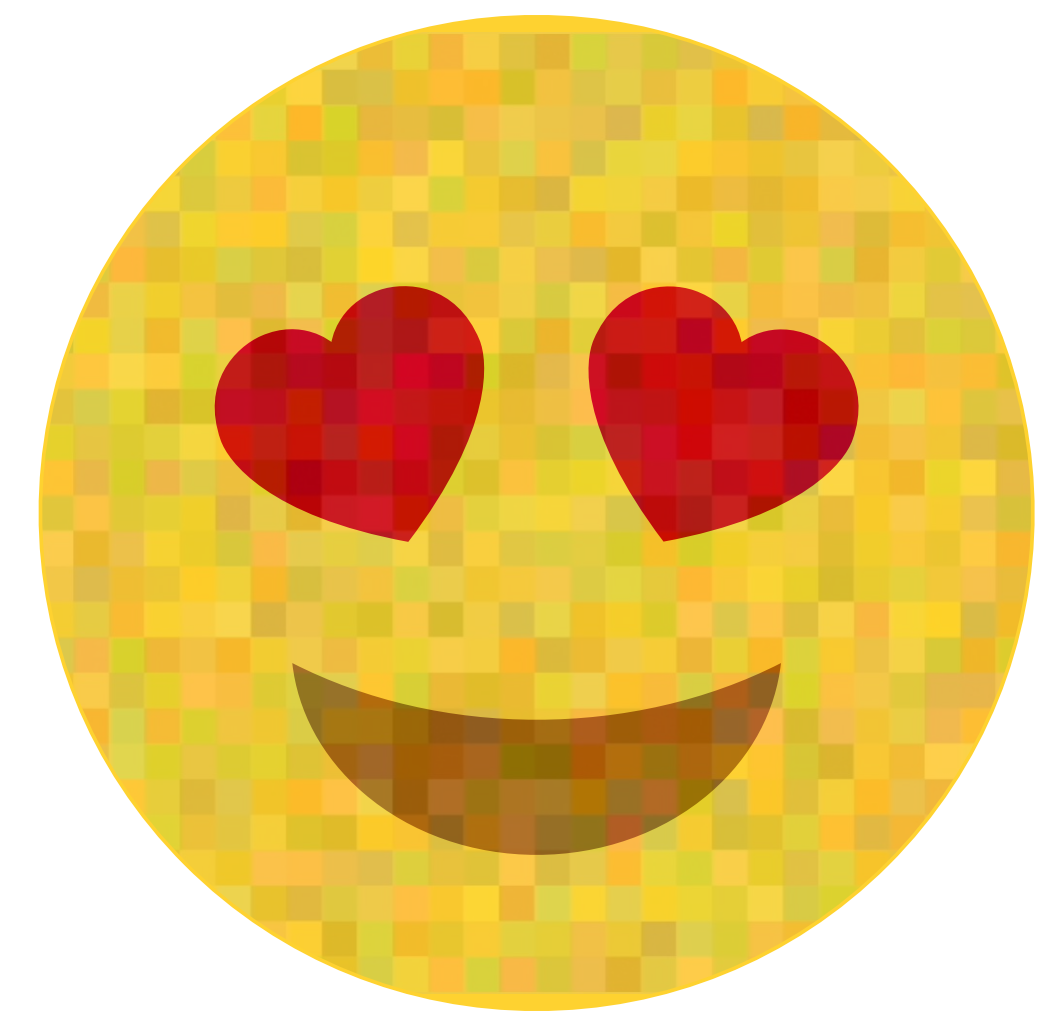
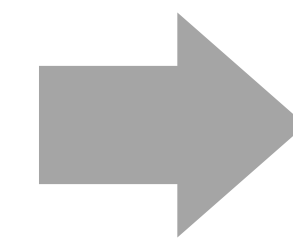
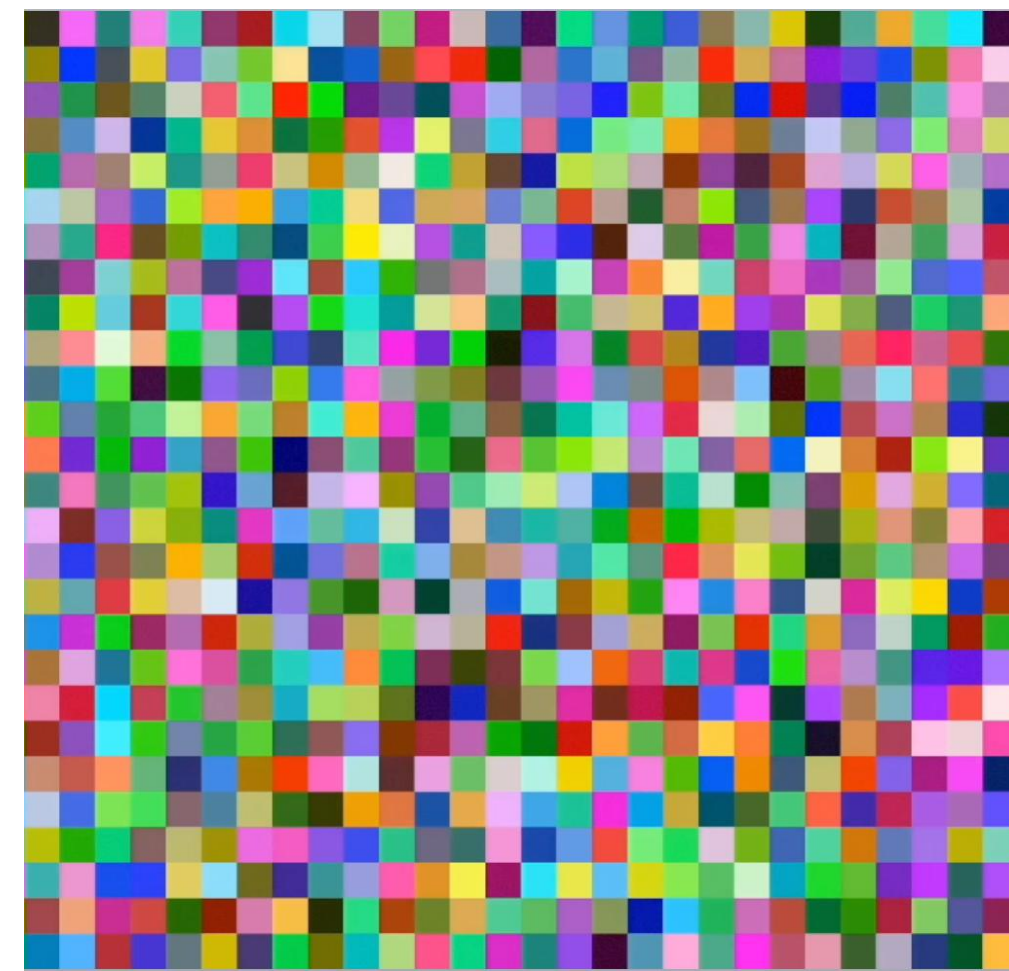
Szegedy et al., “Intriguing properties of neural networks.” In *ICLR '14*.



“face with heart eyes emoji”

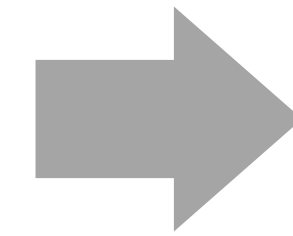
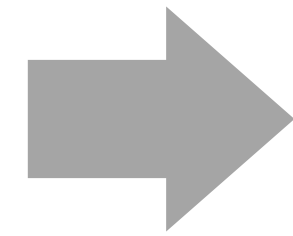
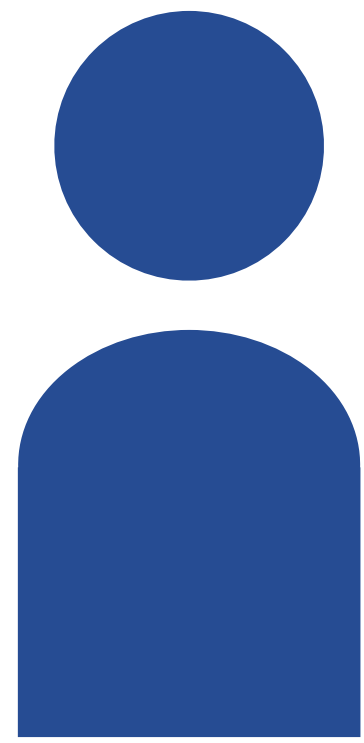


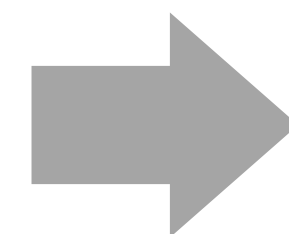
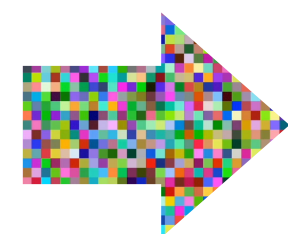
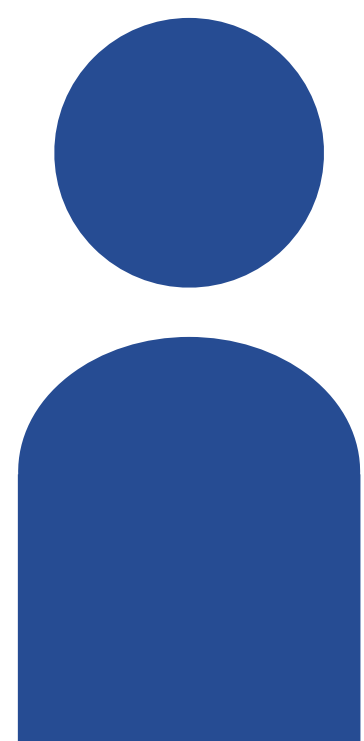
“face with heart eyes emoji”

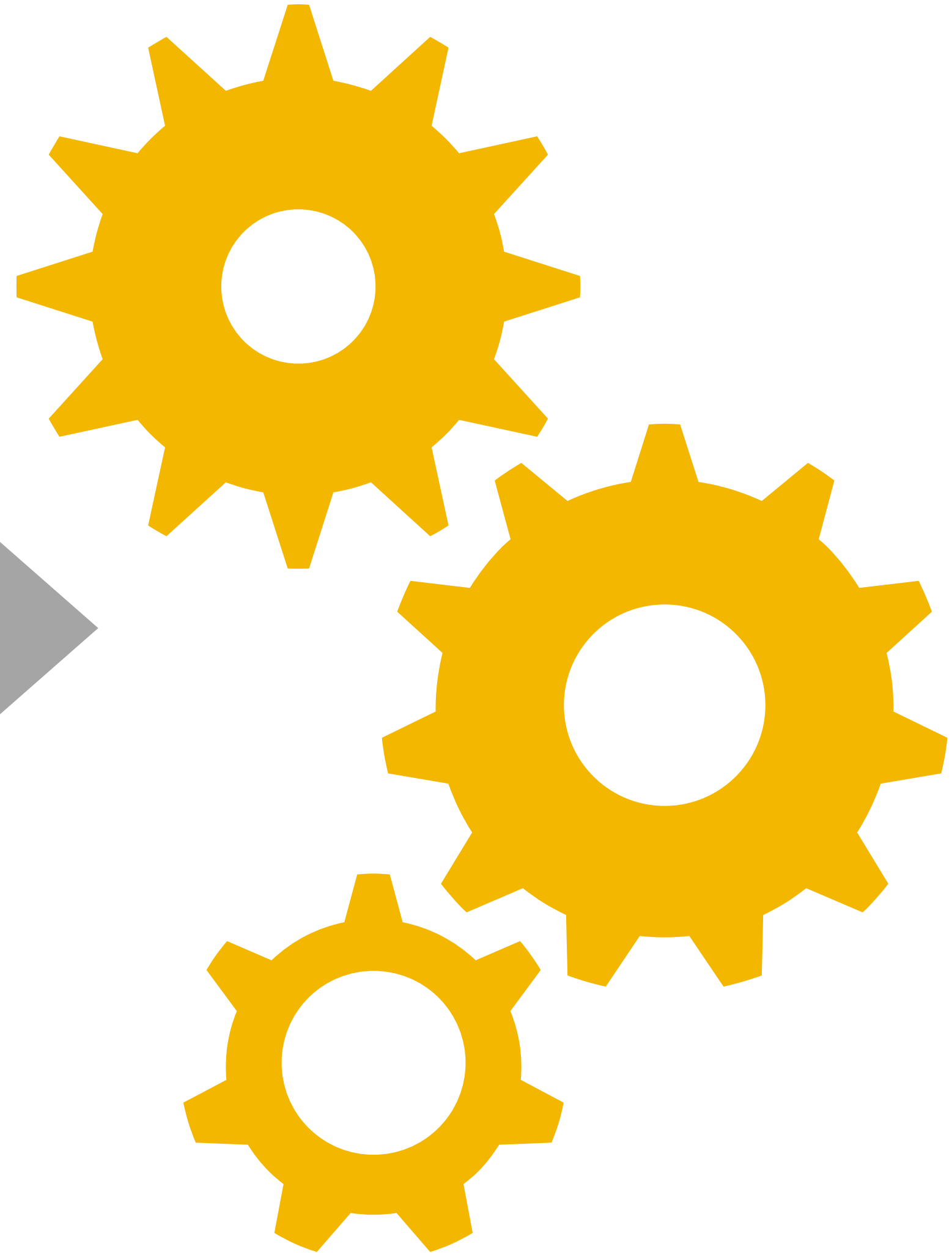
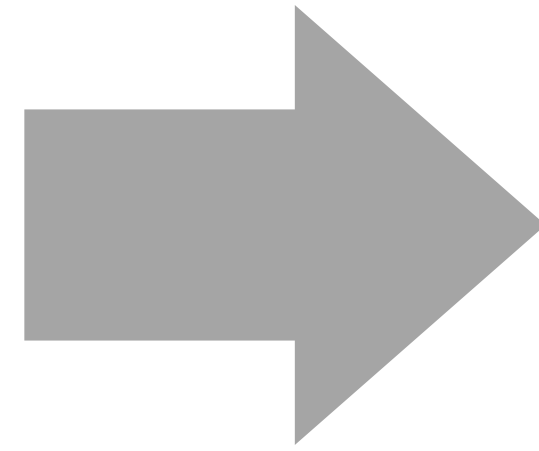
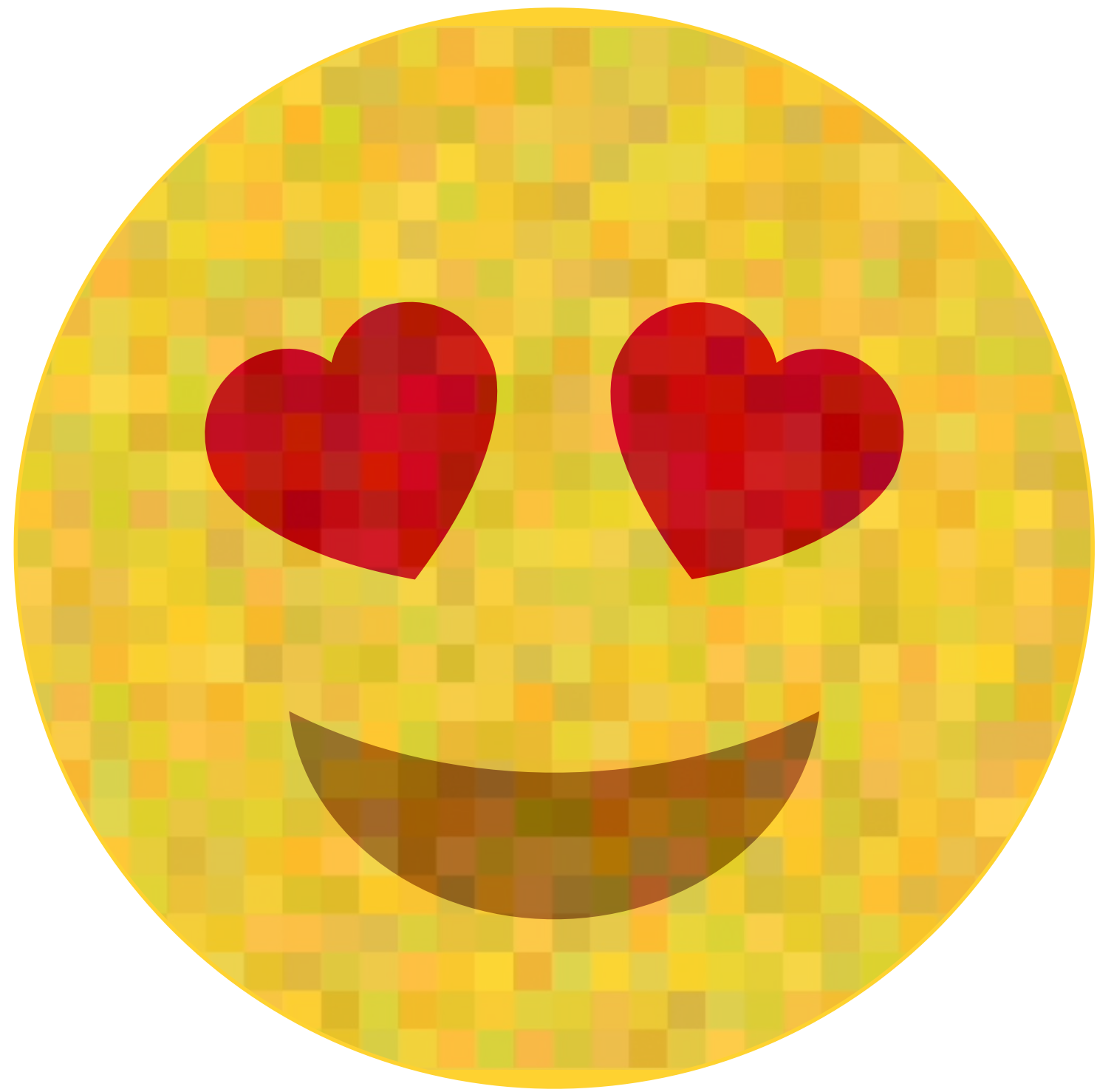


“face with heart eyes emoji”

“stack of pancakes emoji”







We report empirical results and variance numbers for several hundred experimental runs, corresponding to over 250,000 GPU hours on the standard WMT English to German translation task.



Britz et al., “Massive Exploration of Neural Machine Translation Architectures.” In *EMNLP* ’17.

We report empirical results and variance numbers for several hundred experimental runs, corresponding to over 250,000 GPU hours on the standard WMT English to German translation task.



Britz et al., “Massive Exploration of Neural Machine Translation Architectures.” In *EMNLP* ’17.

We report empirical results and variance numbers for several hundred experimental runs, corresponding to over 250,000 GPU hours on the standard WMT English to German translation task.



Britz et al., “Massive Exploration of Neural Machine Translation Architectures.” In *EMNLP* ’17.



Search

Szilard Pafka

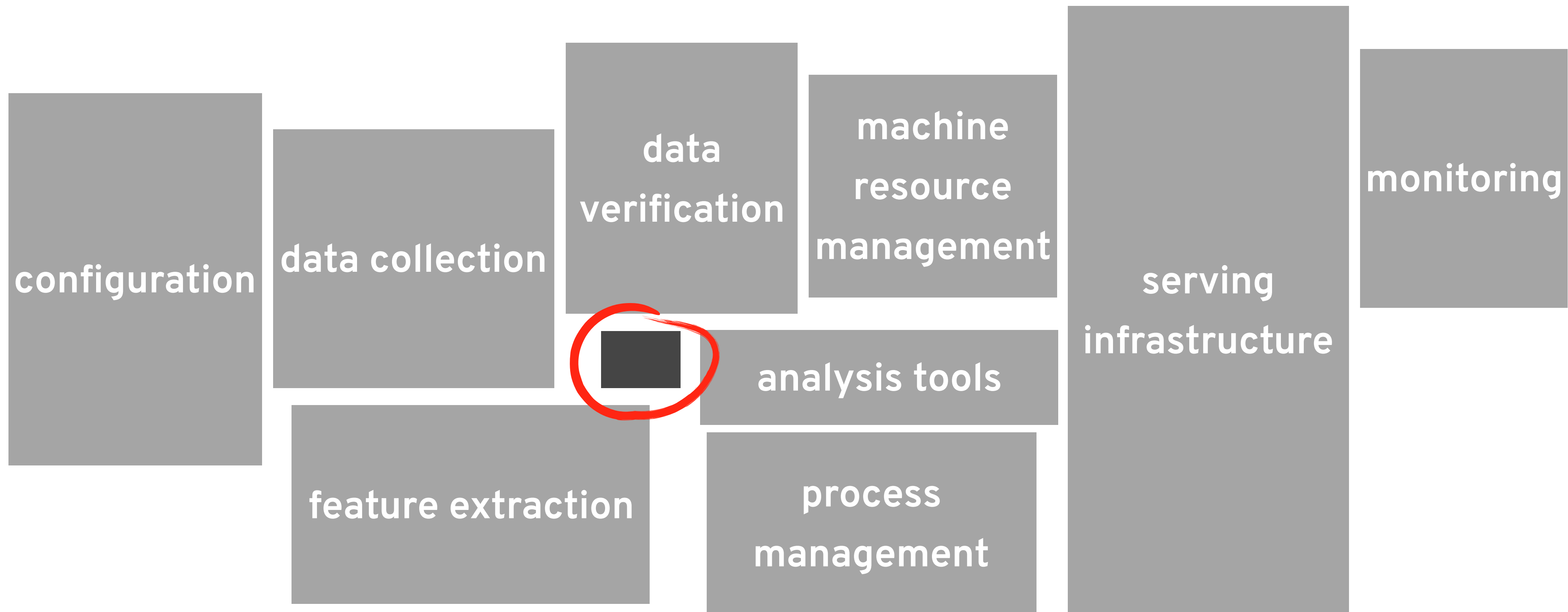
Better than Deep Learning: Gradient Boosting Machines (GBM)

06/17/2019 - 15:20 to 16:00

Palais Atelier

long talk (40 min)





Forecast

What's wrong with machine learning?

The wrong tools

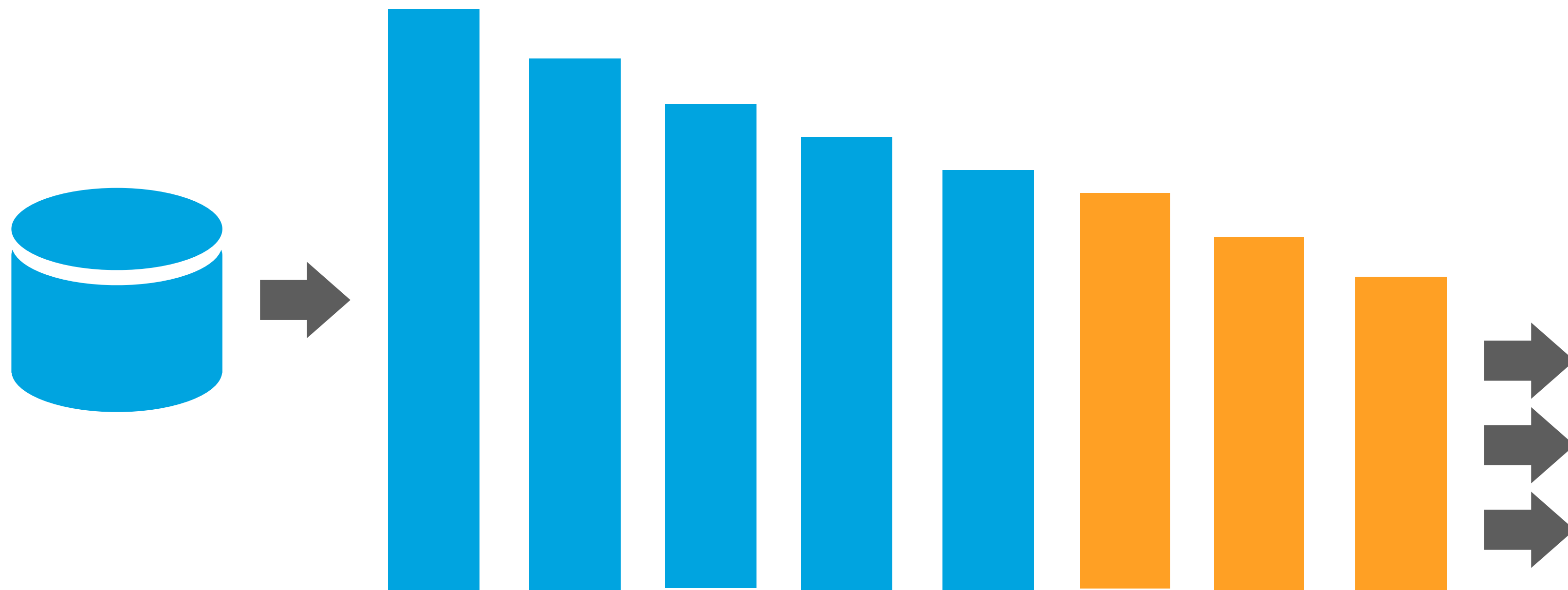
The wrong incentives

The wrong problems

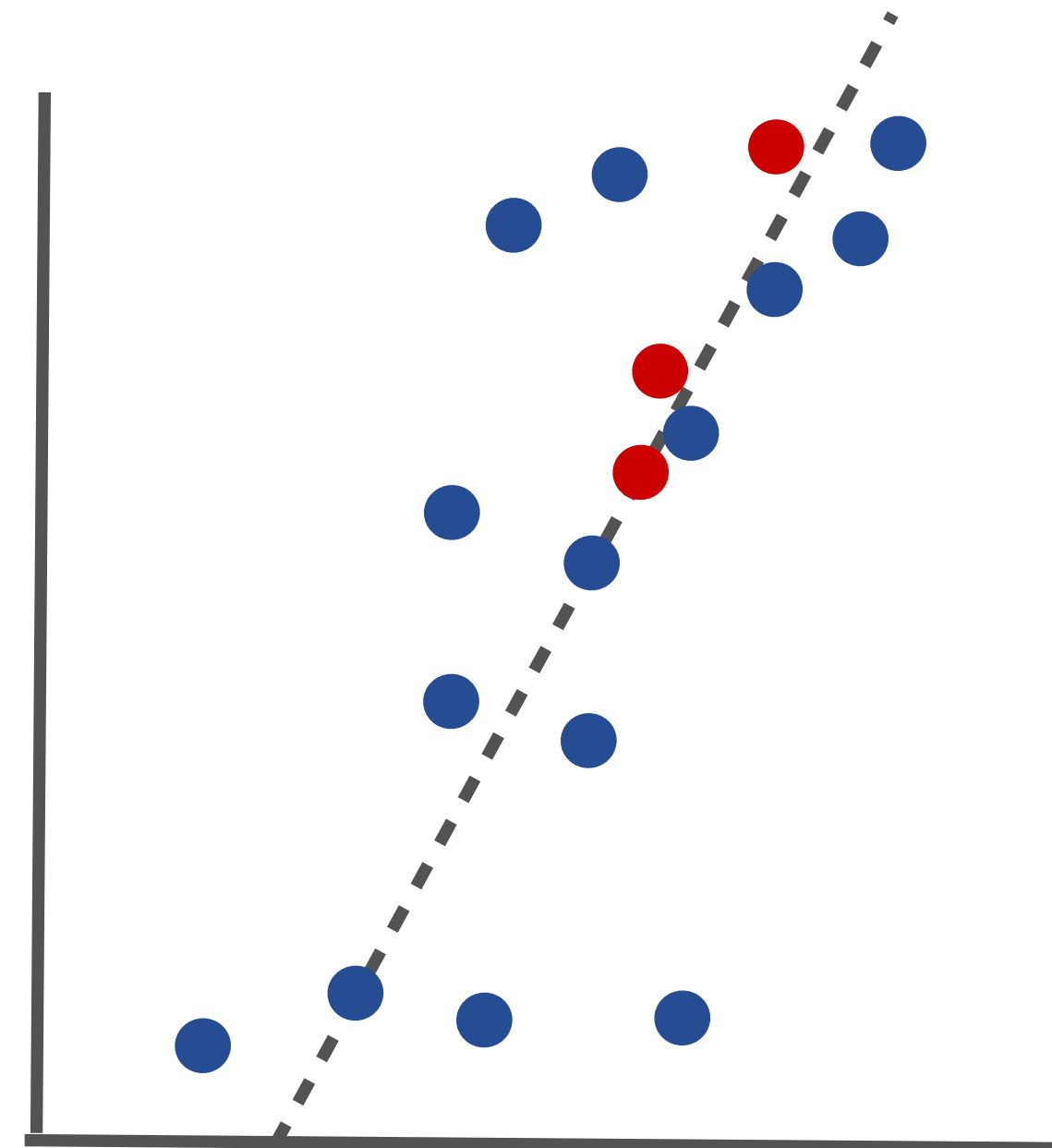
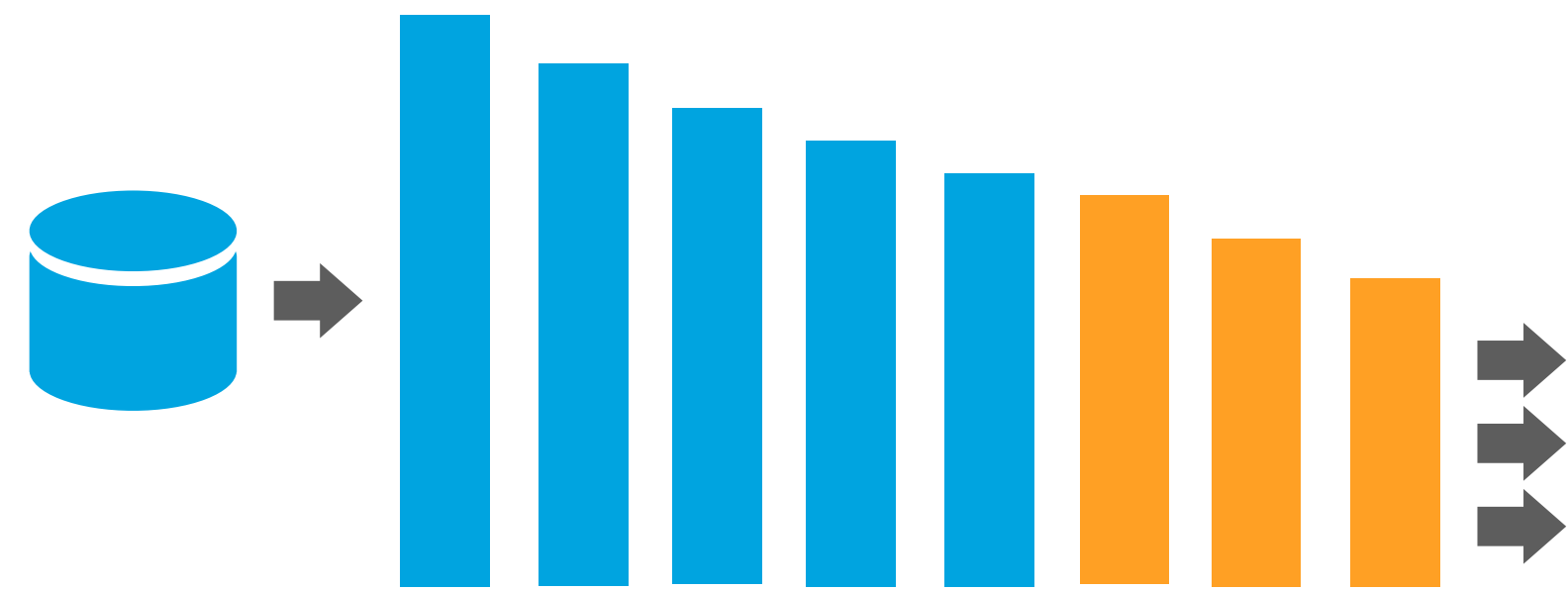
The way forward

We unfairly reward complexity

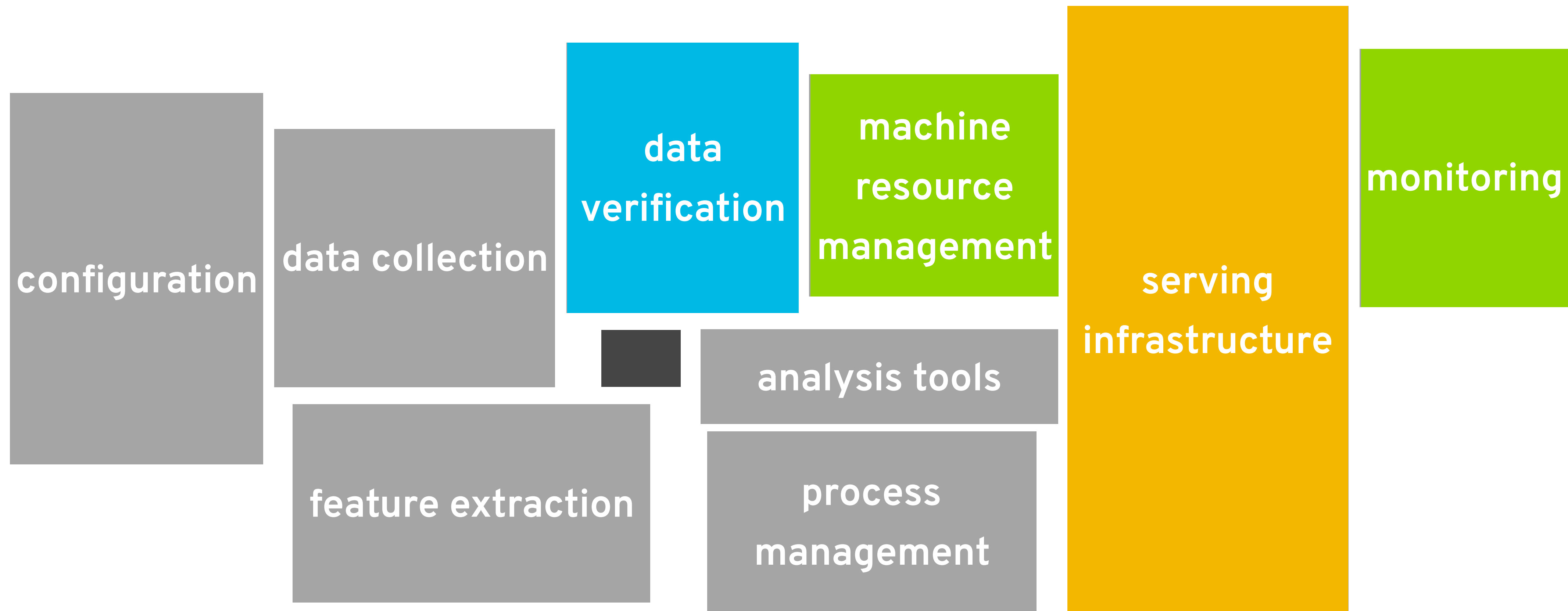
We unfairly reward complexity

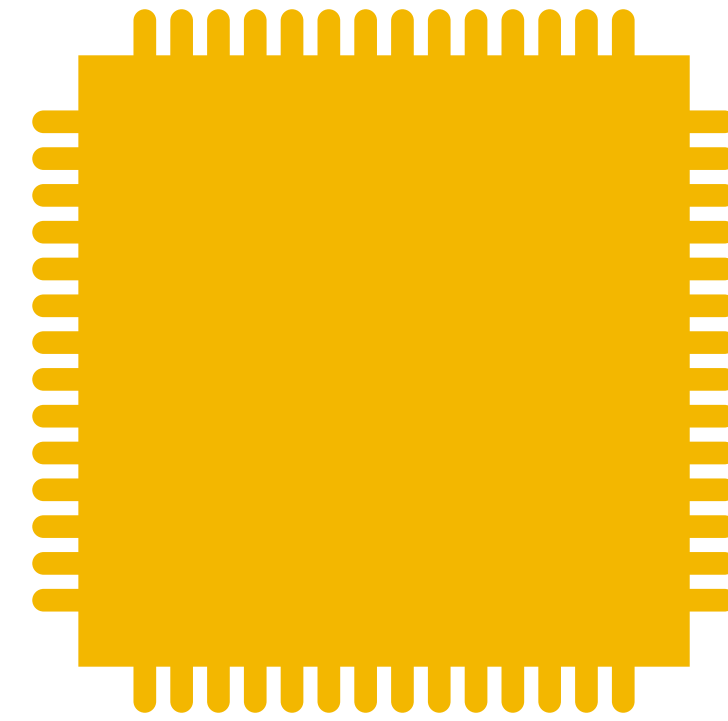
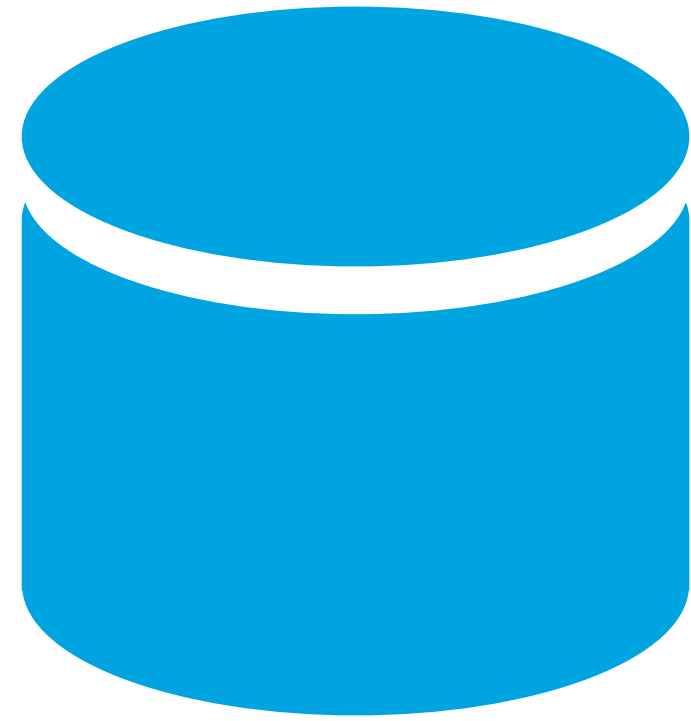


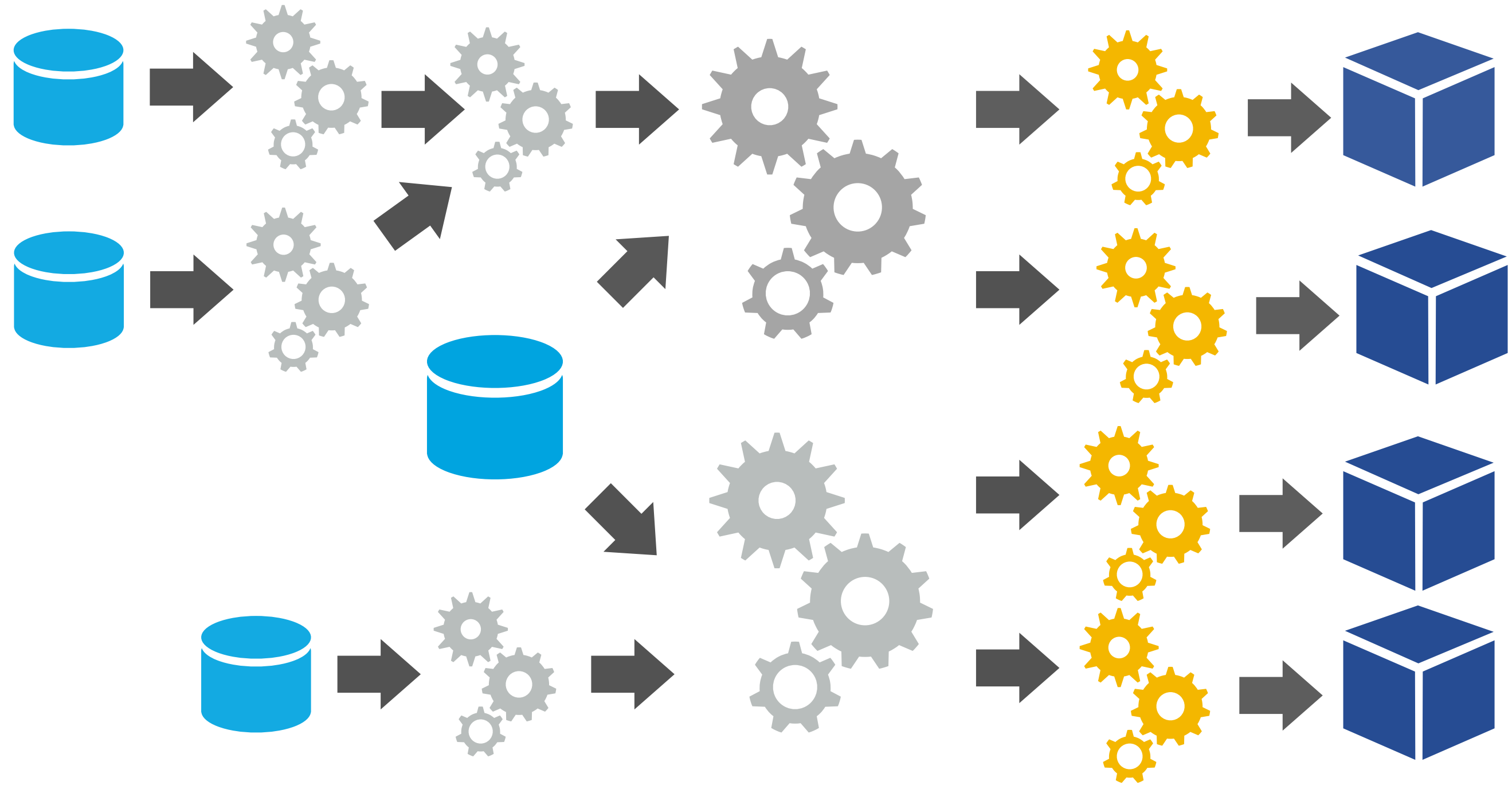
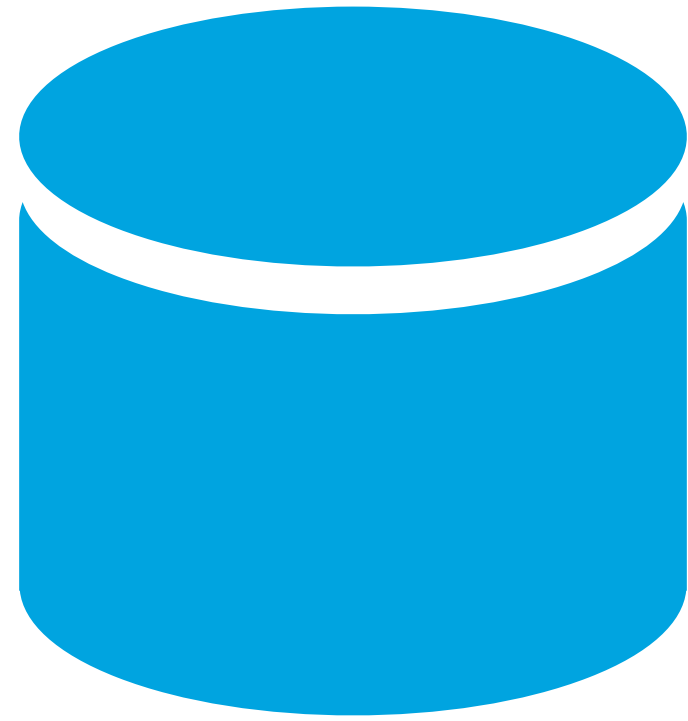
We unfairly reward complexity

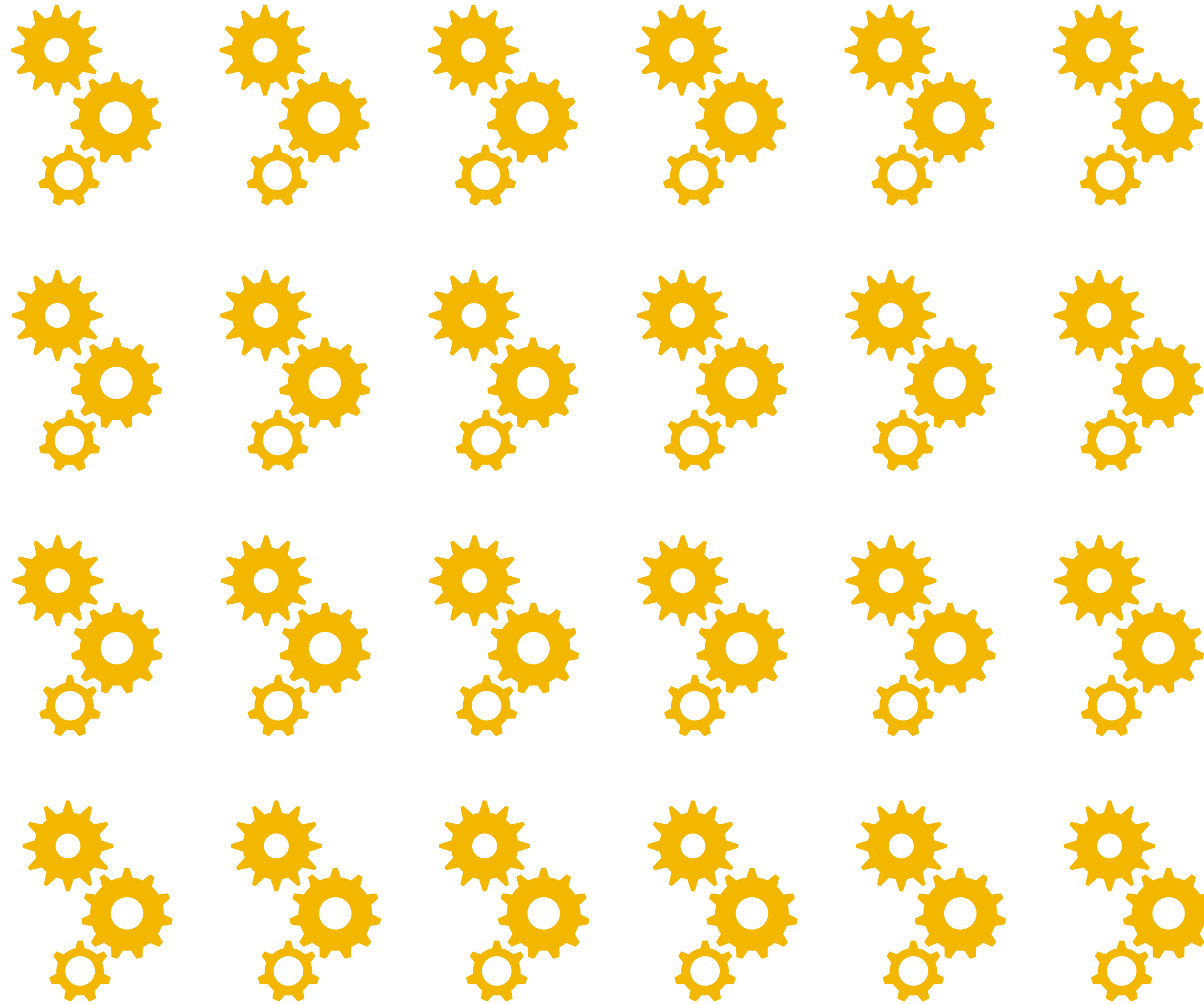


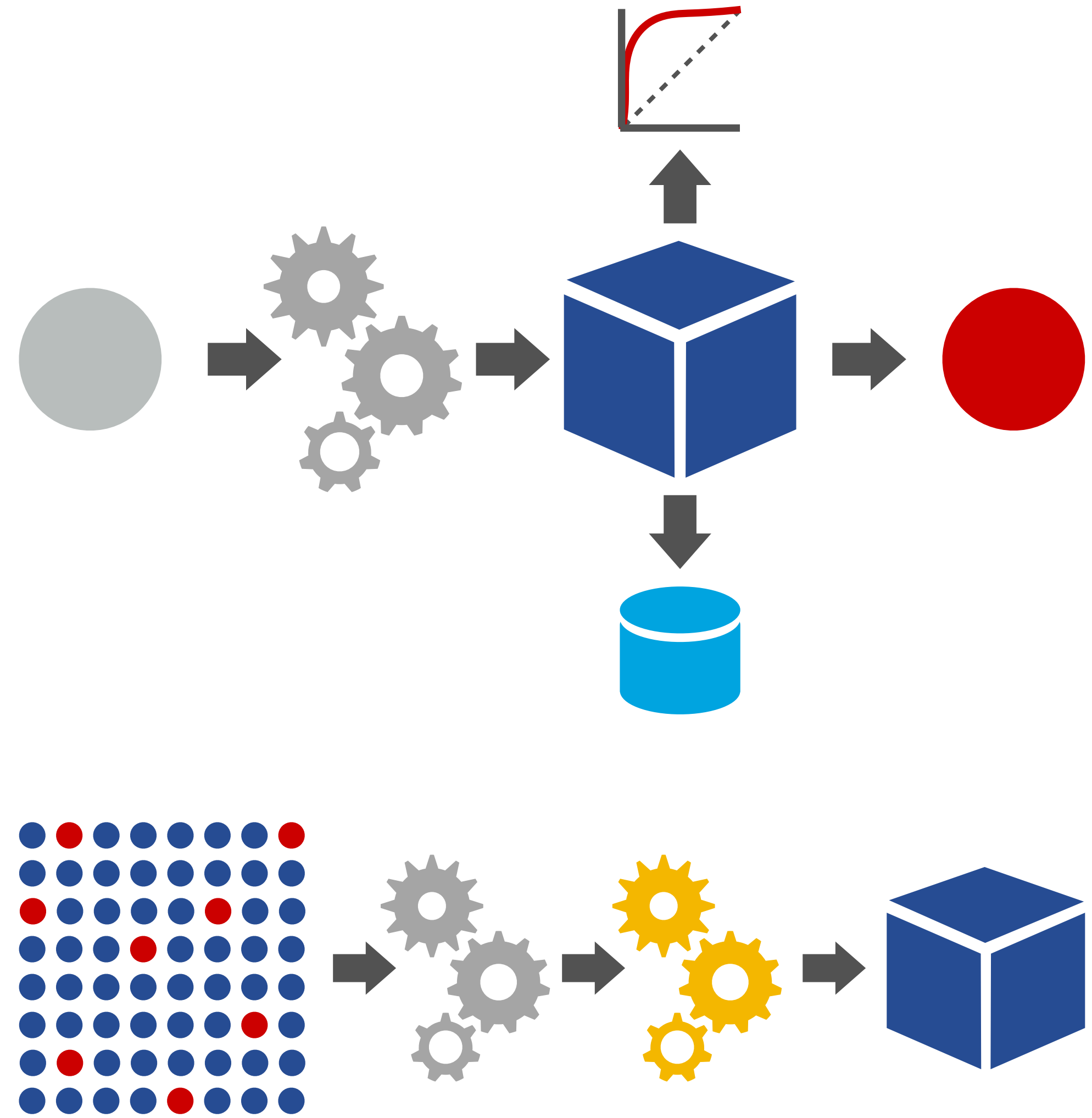
Vendor incentives

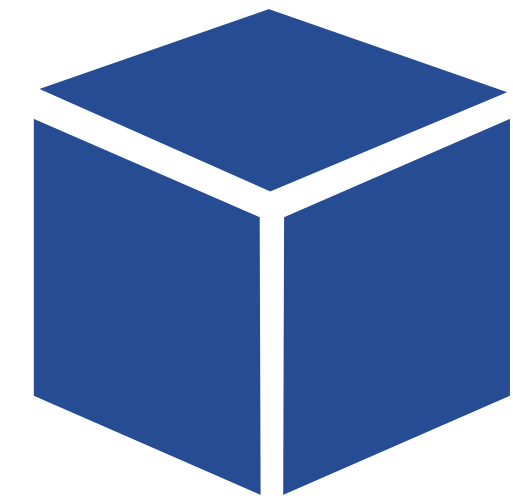
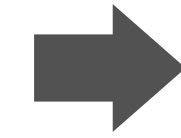
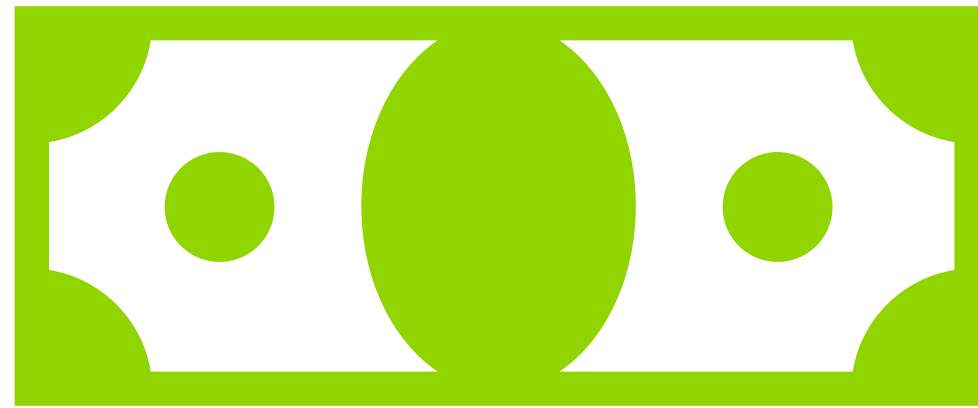


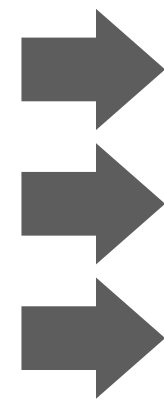
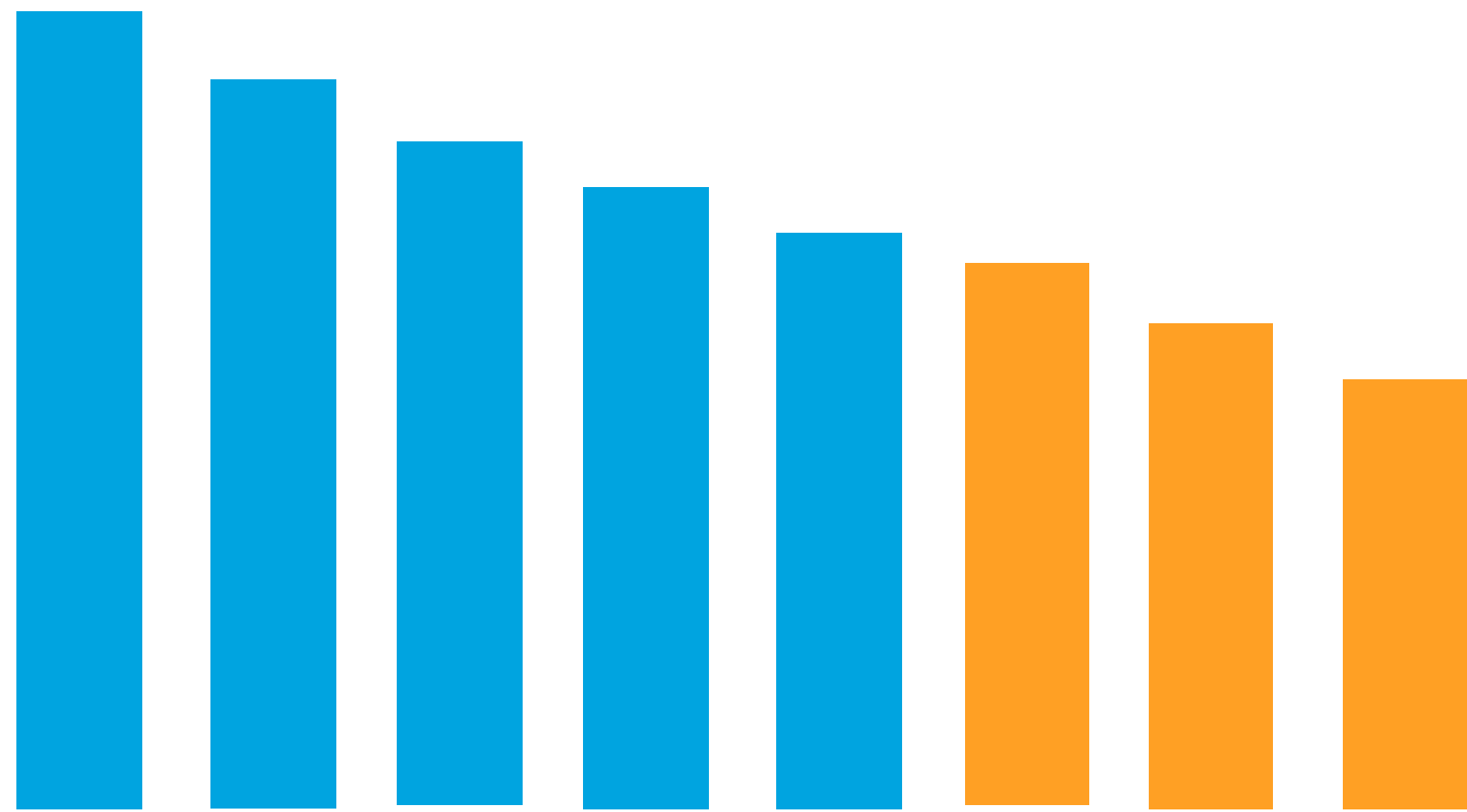
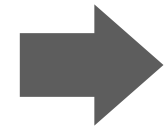
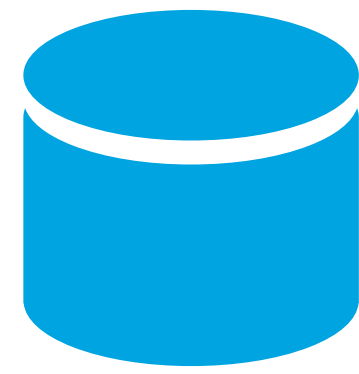
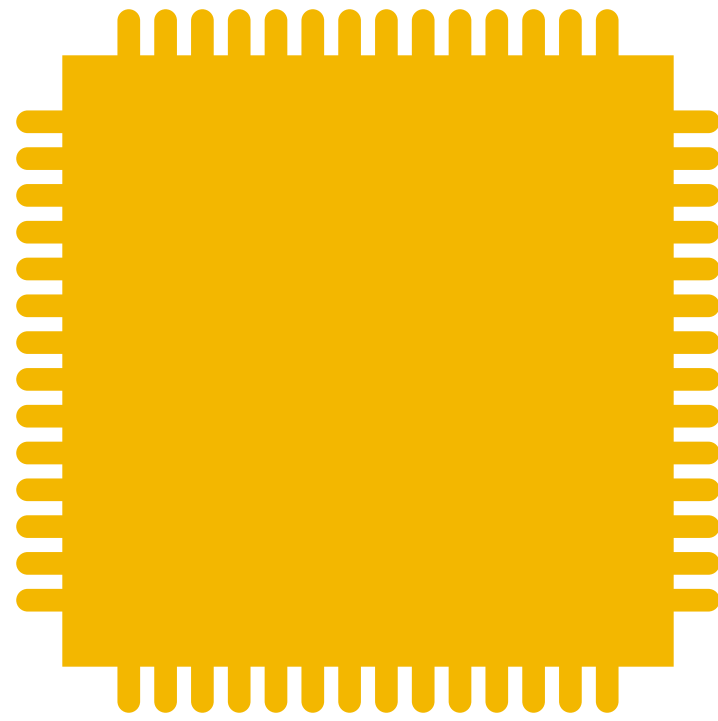




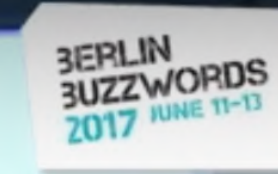
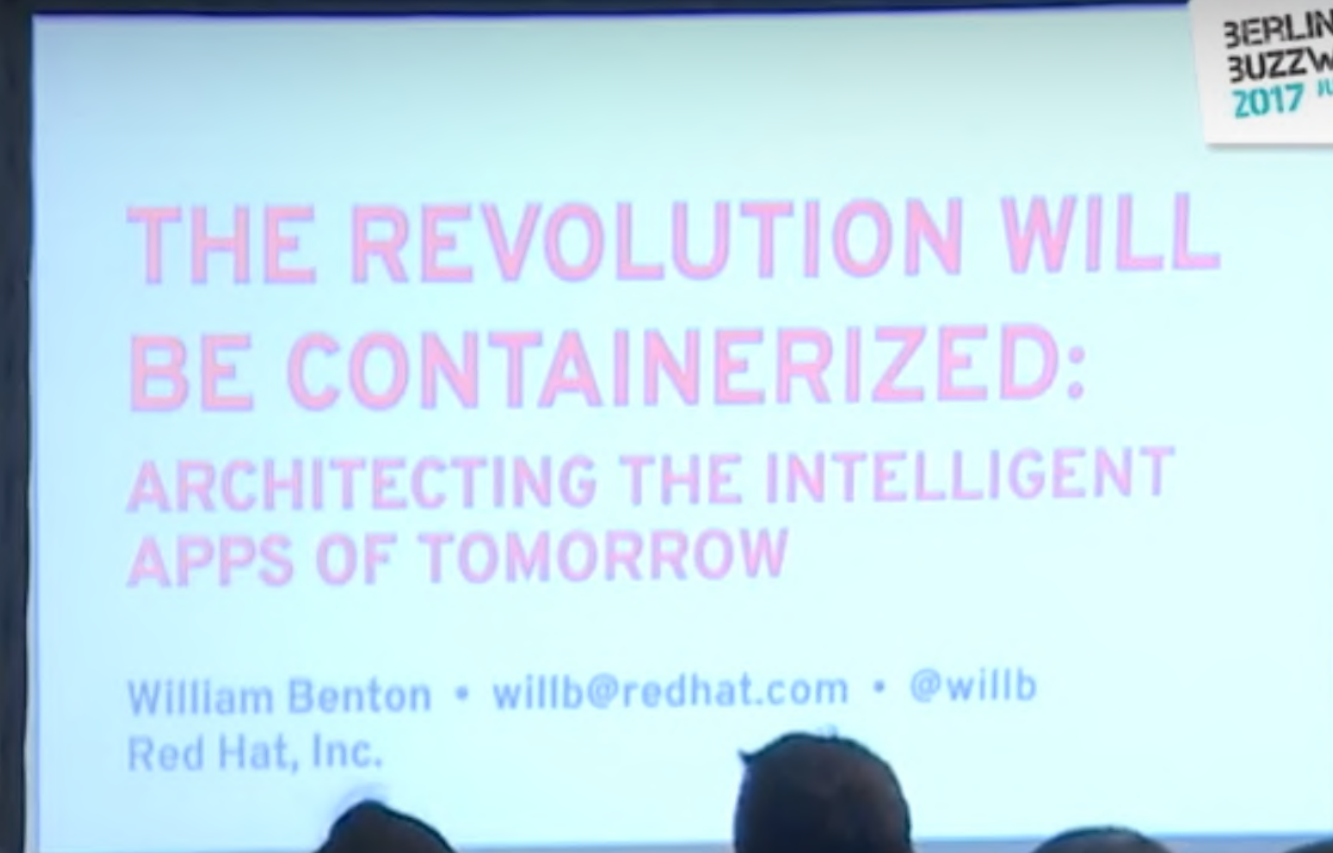






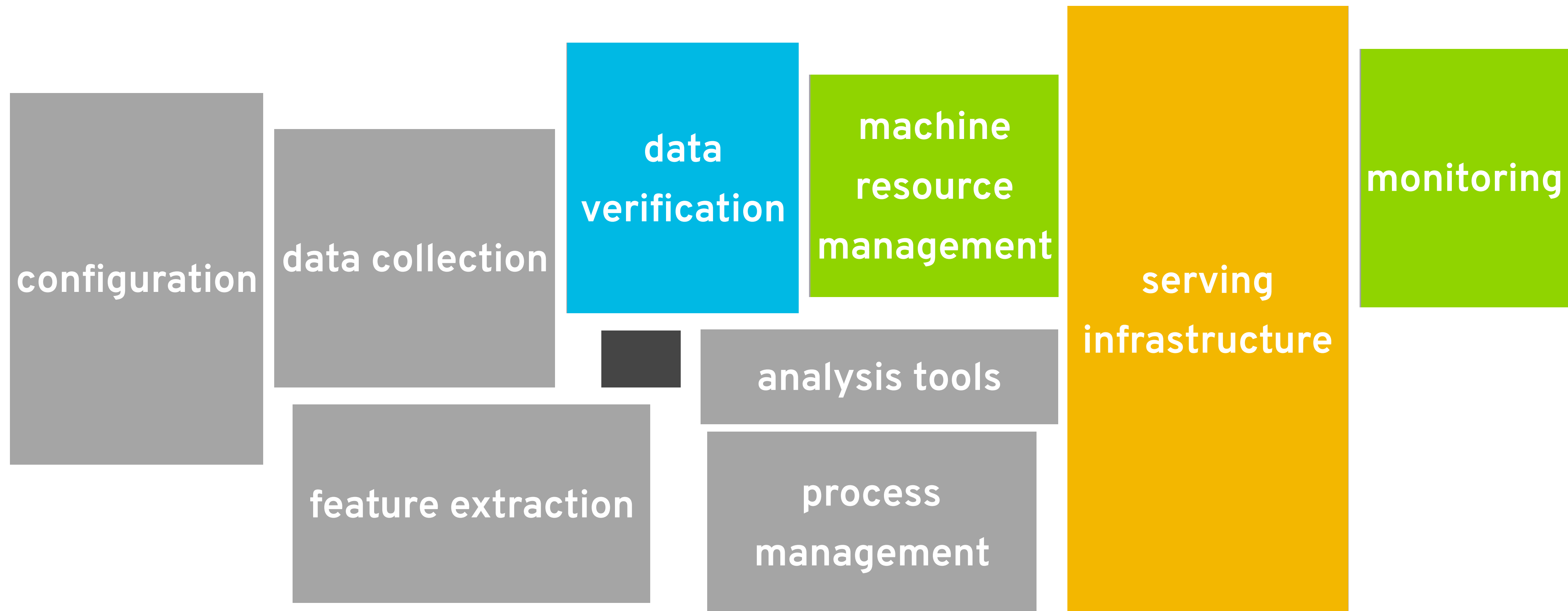


#bbuzz 17: William Benton-The Revolution will be containerized



@willb #bbuzz





monitoring

NOW WE'VE GOT
PROBLEMS AND I
DON'T THINK WE
CAN **SOLVE THEM**

configur

feature e

ment

Forecast

What's wrong with machine learning?

The wrong tools

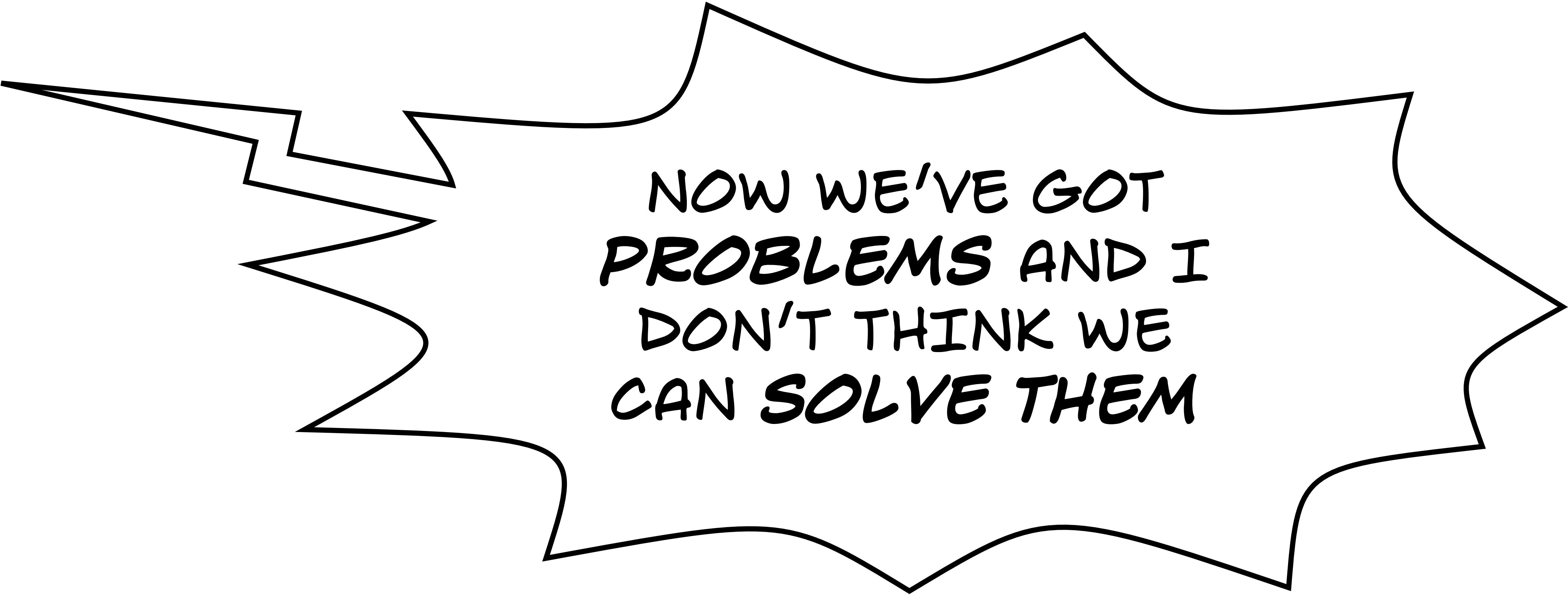
The wrong incentives

The wrong problems

The way forward

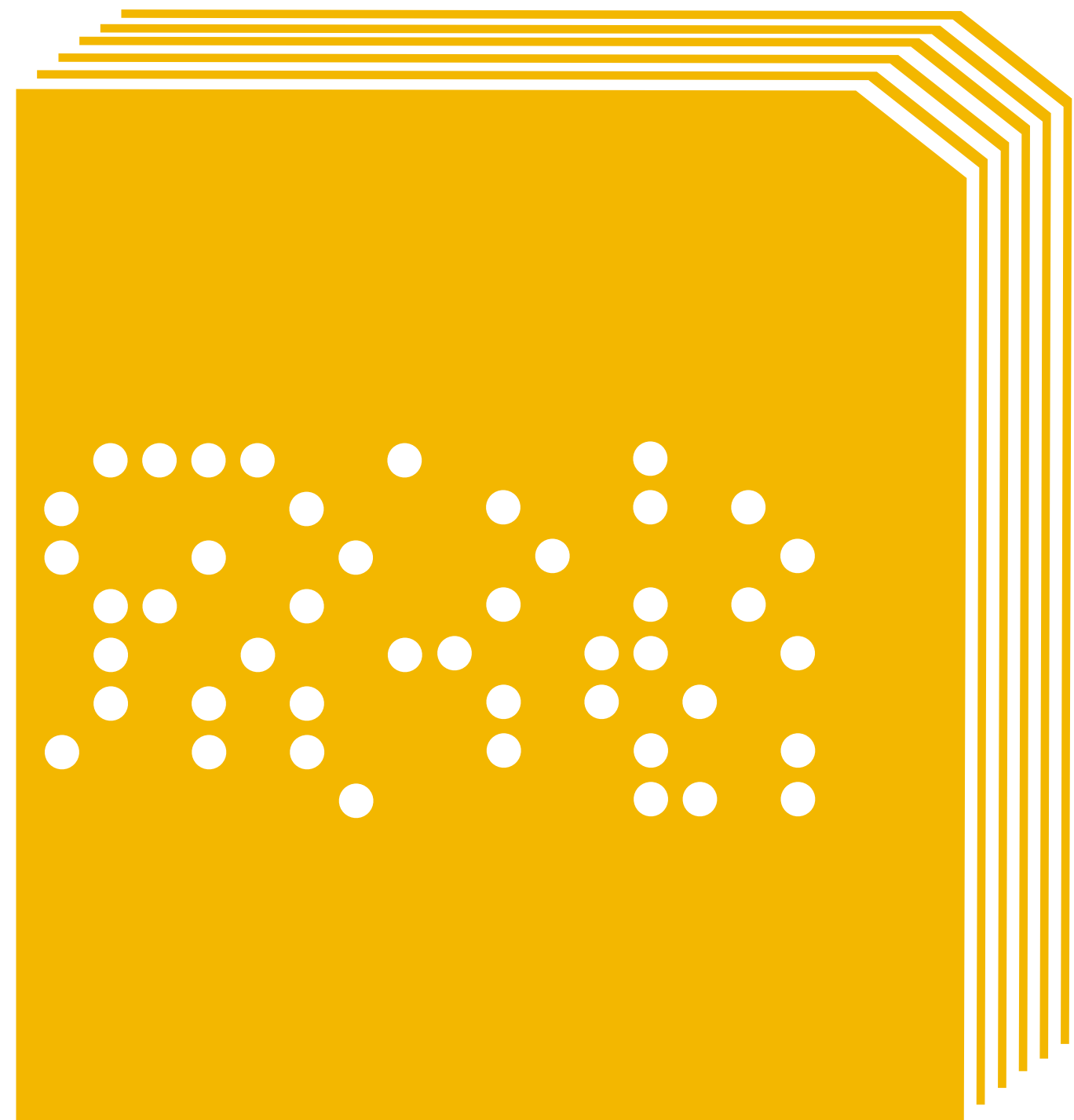
Programming is one of the most difficult branches of applied mathematics; the poorer mathematicians had better remain pure mathematicians.

– *Edsger Dijkstra, 18 June 1975*

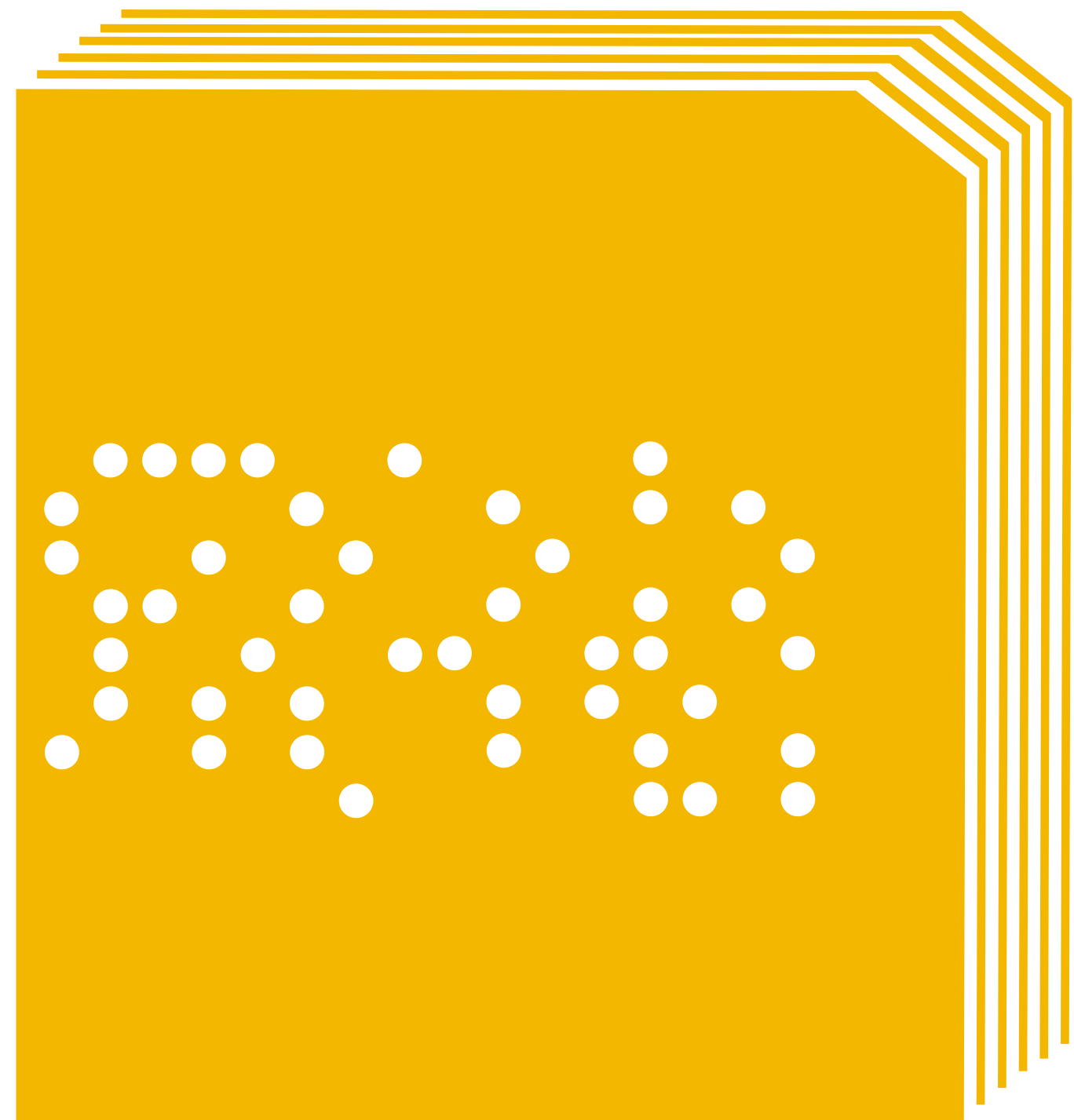


NOW WE'VE GOT
PROBLEMS AND I
DON'T THINK WE
CAN **SOLVE THEM**

Interfaces for observability

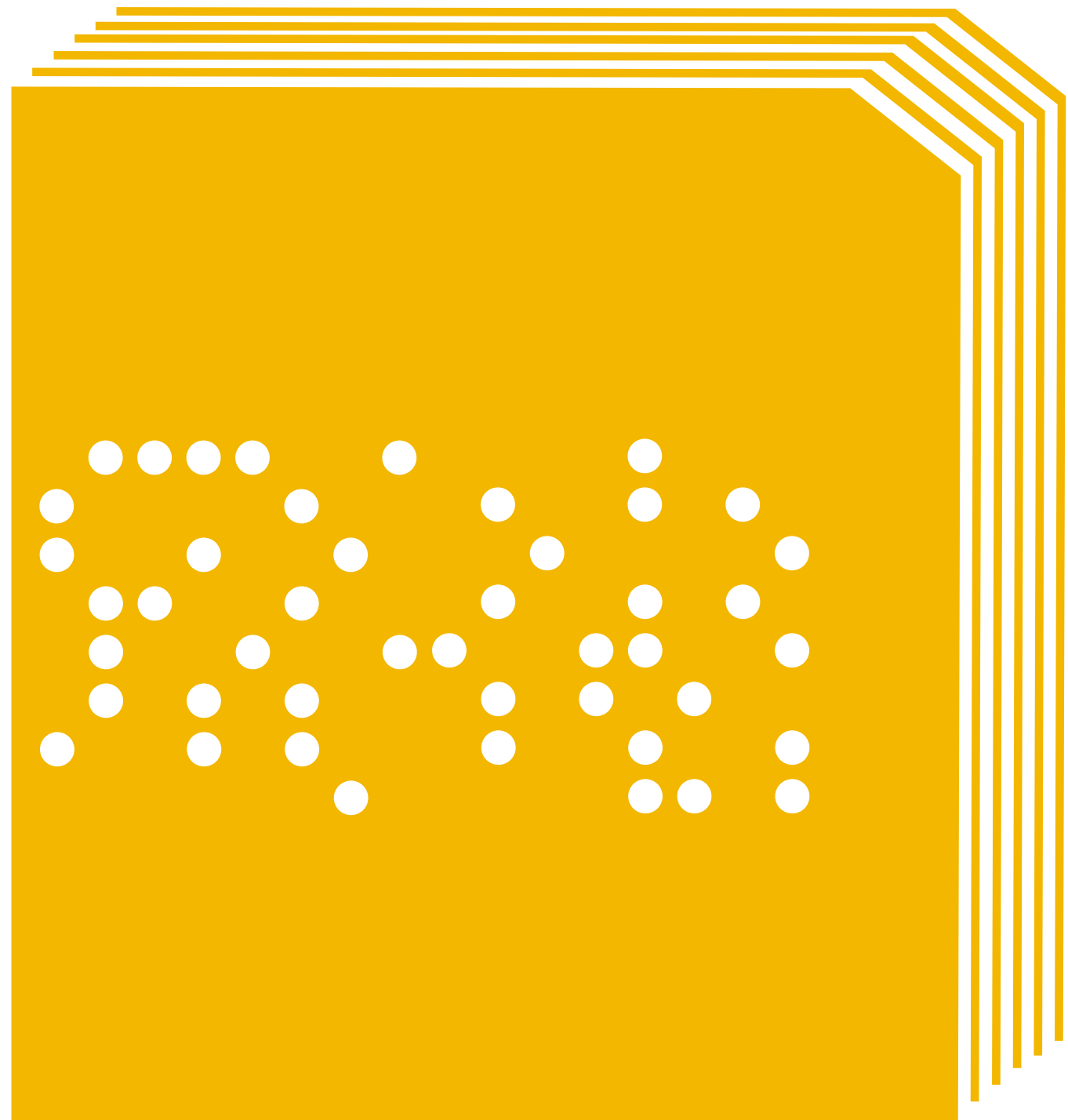


Interfaces for observability



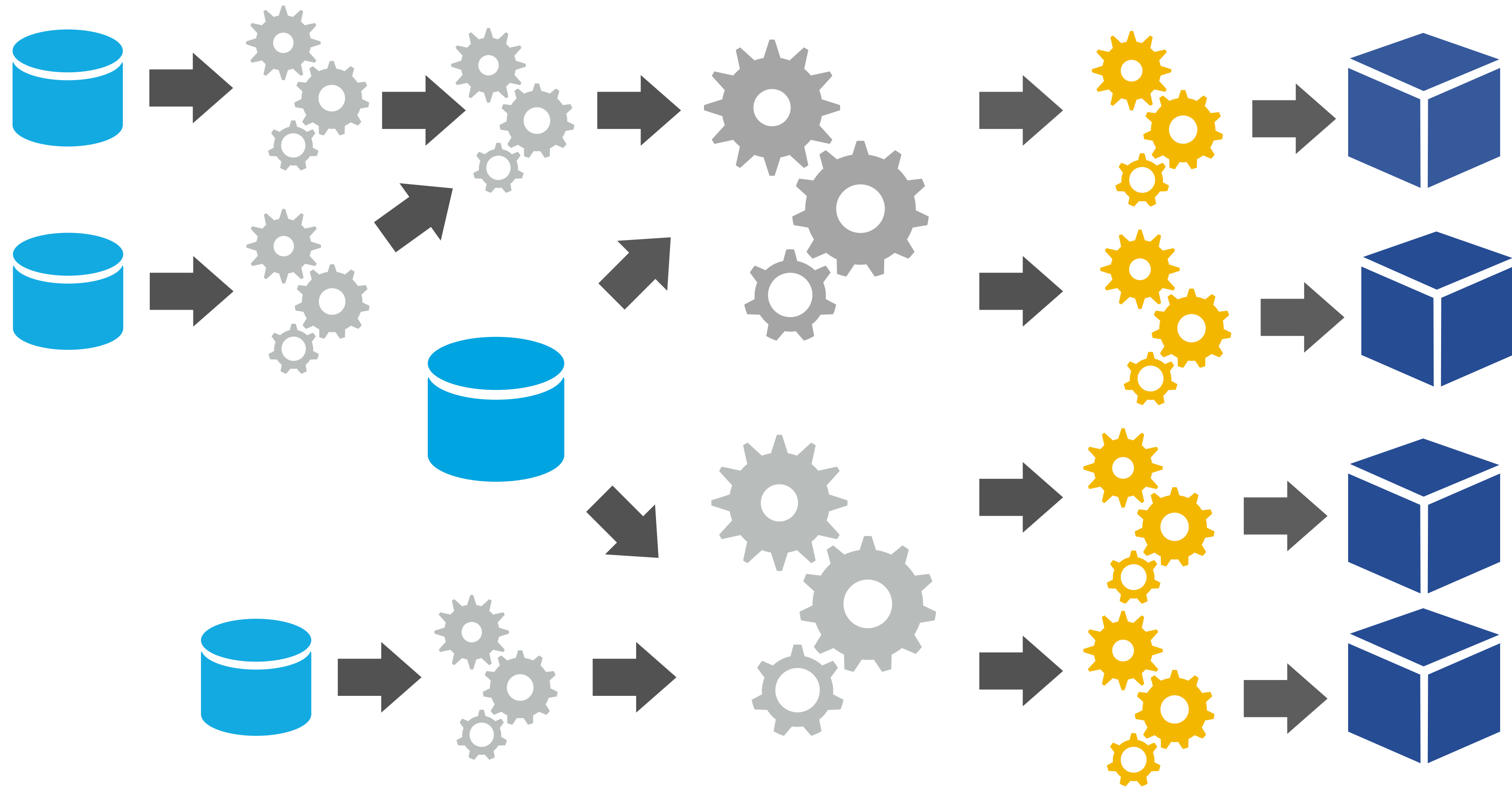
```
gcc -c dot.c
```

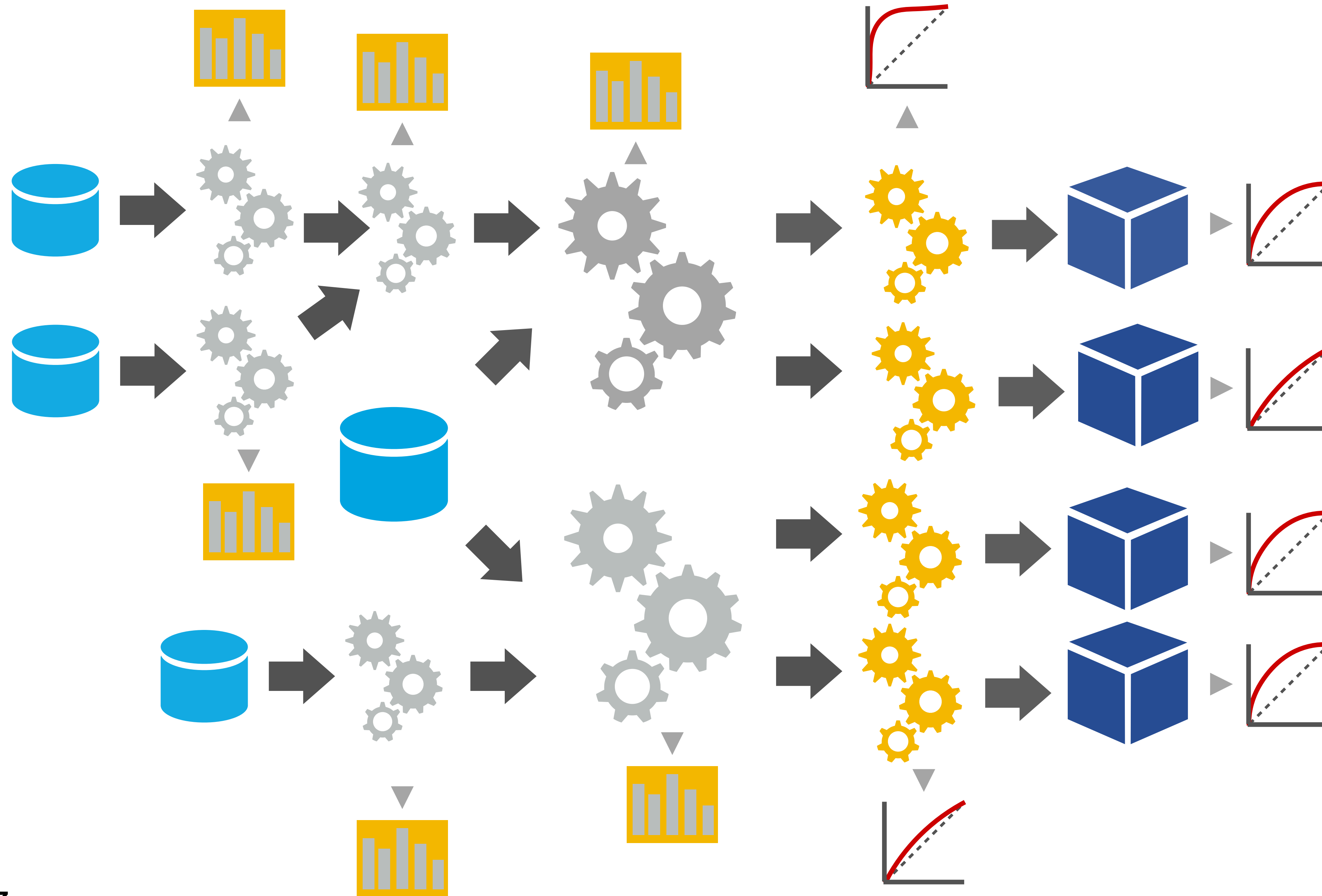
Interfaces for observability



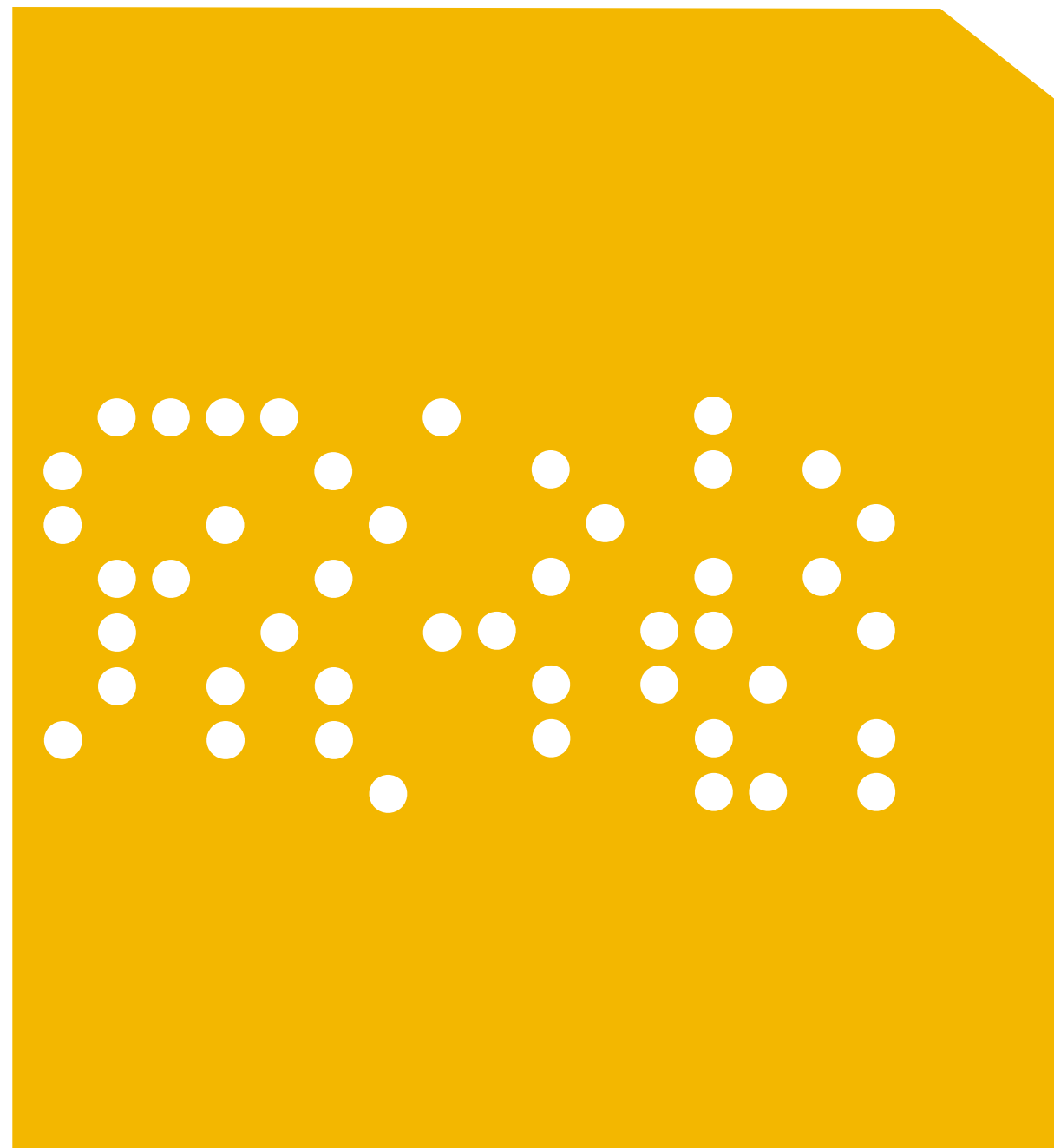
NameError: Name 'Sum' is not defined

```
def dot(xs, ys):  
    return Sum([x * y for  
                x, y in zip(xs, ys)])
```



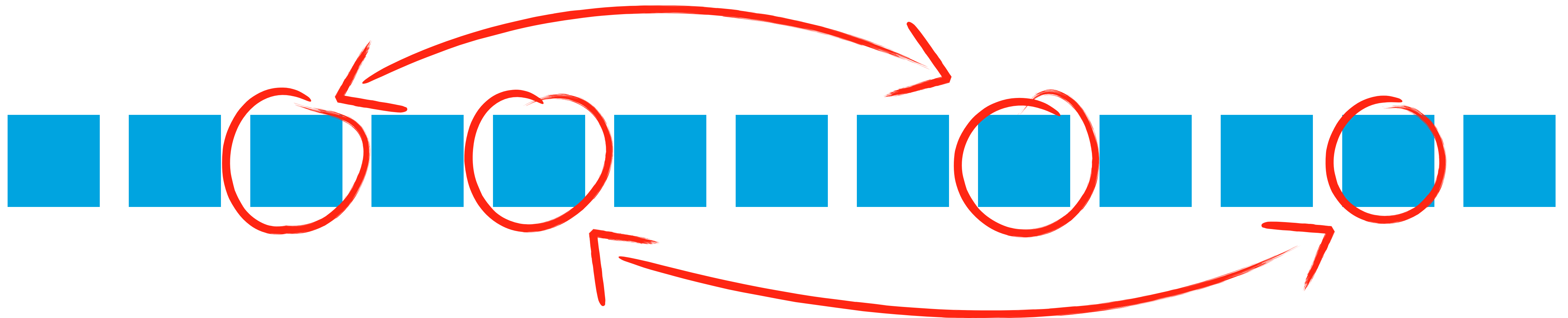


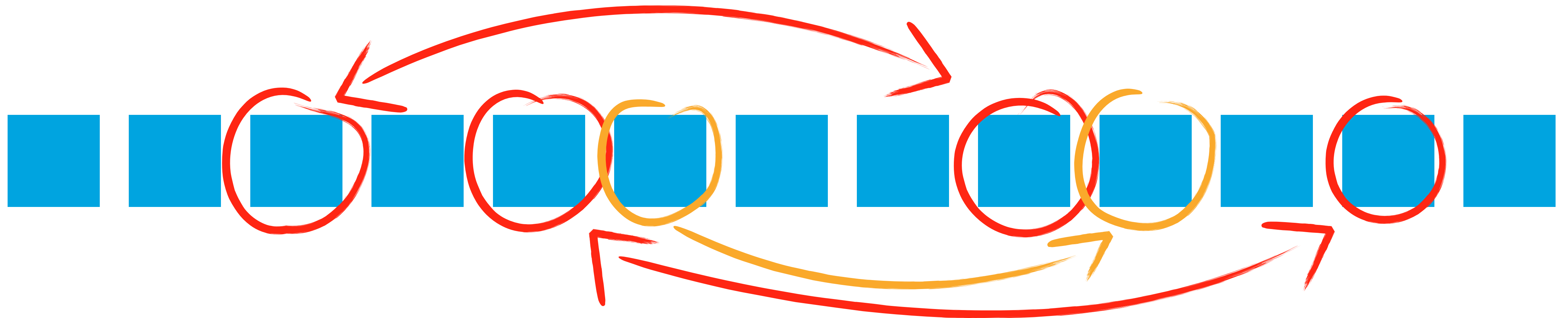
@willb #bbuzz

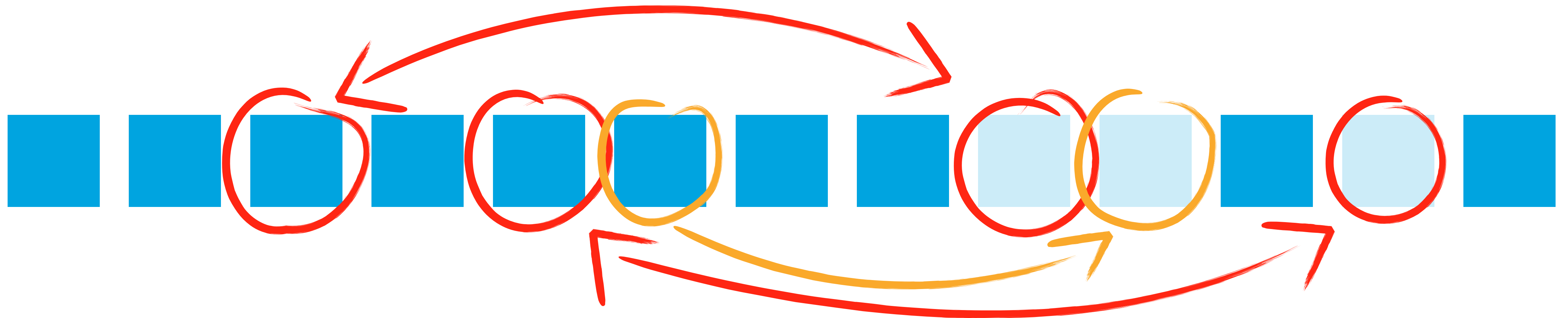


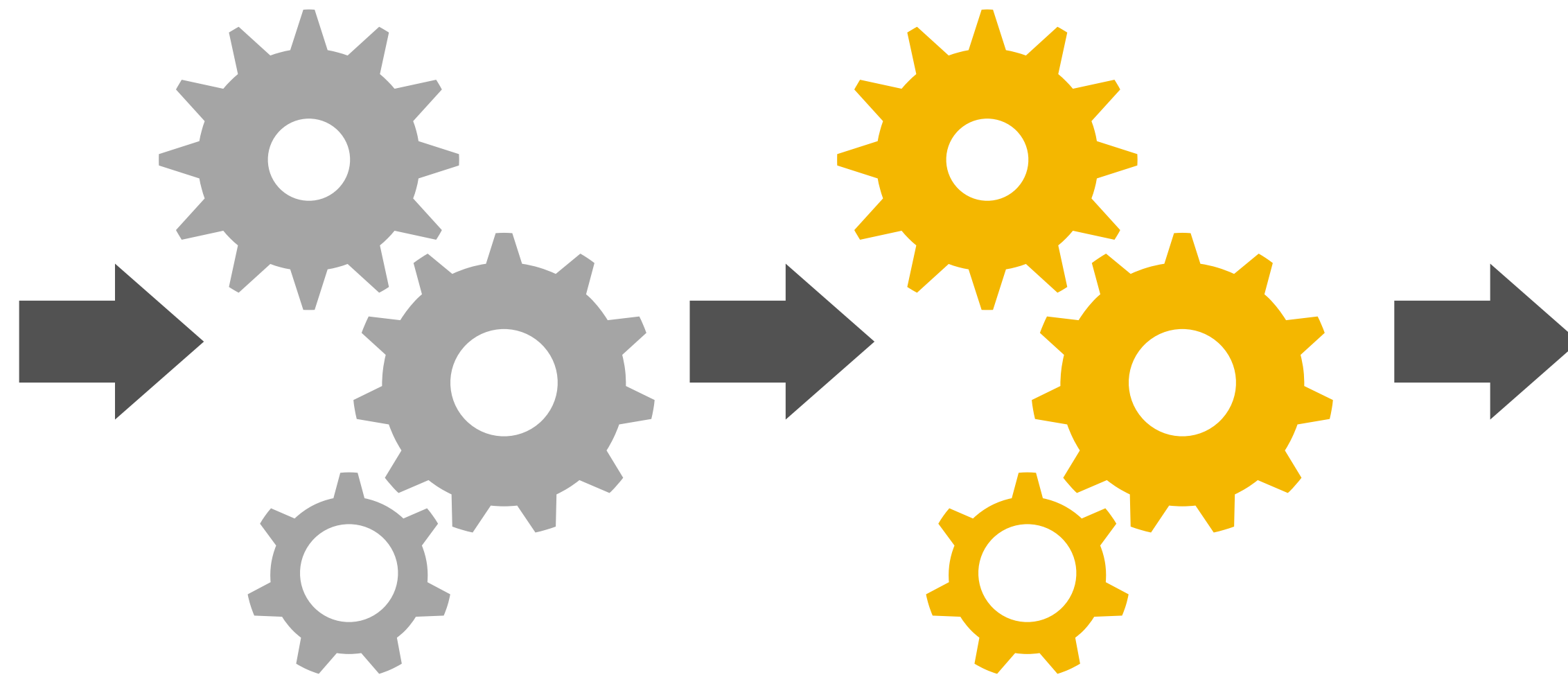
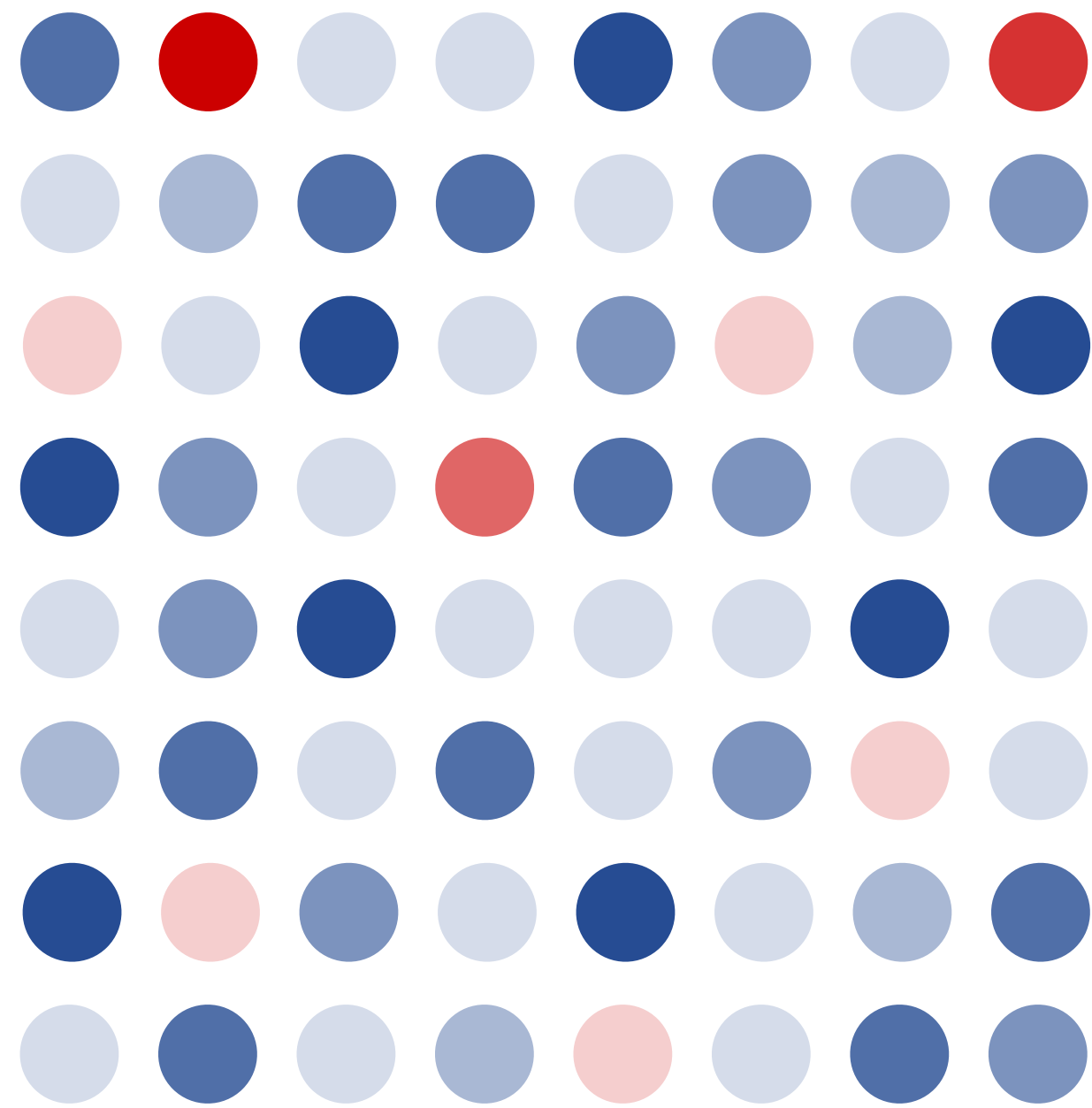
```
def dot(xs, ys):  
    return sum([x * y for  
                x, y in zip(xs, ys)])
```



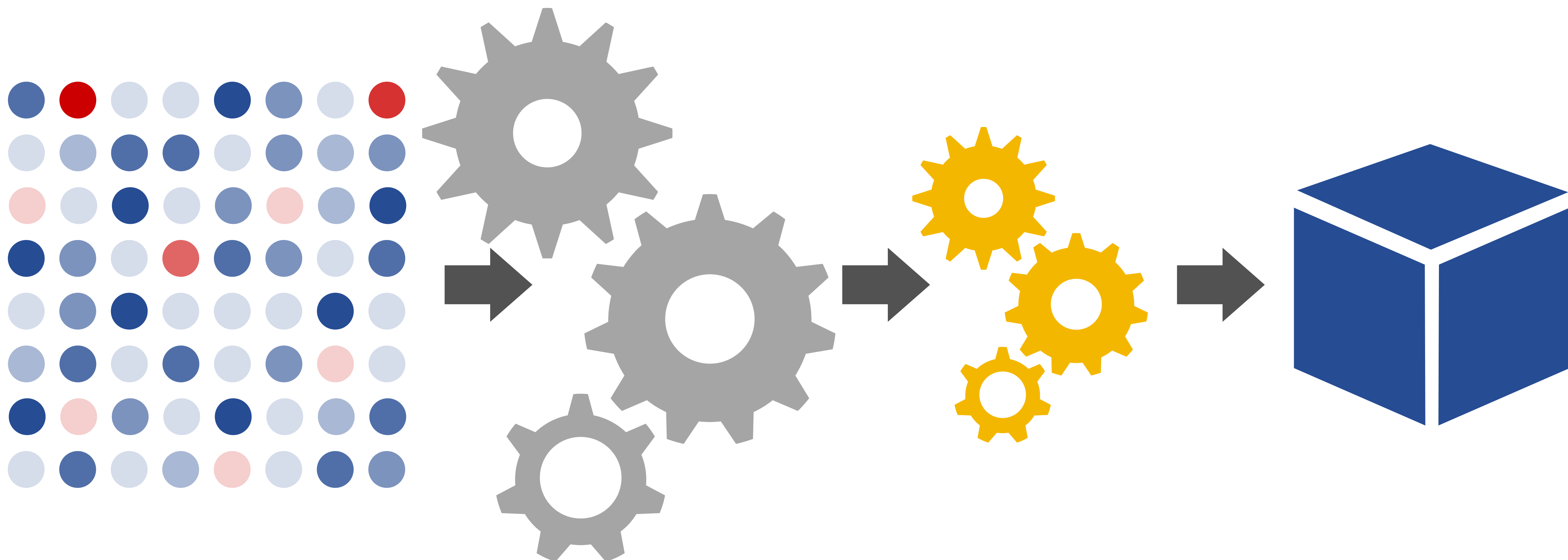




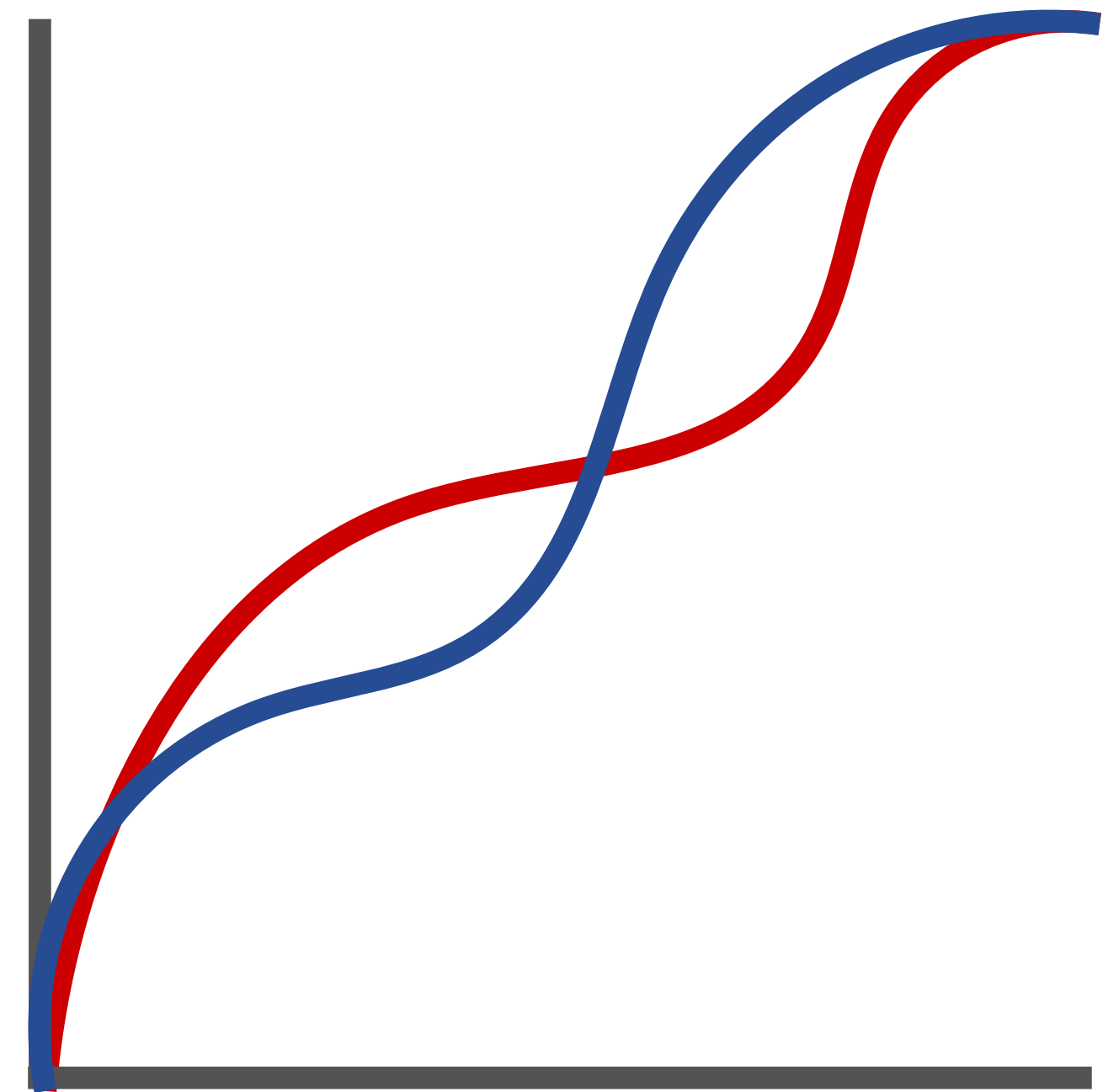
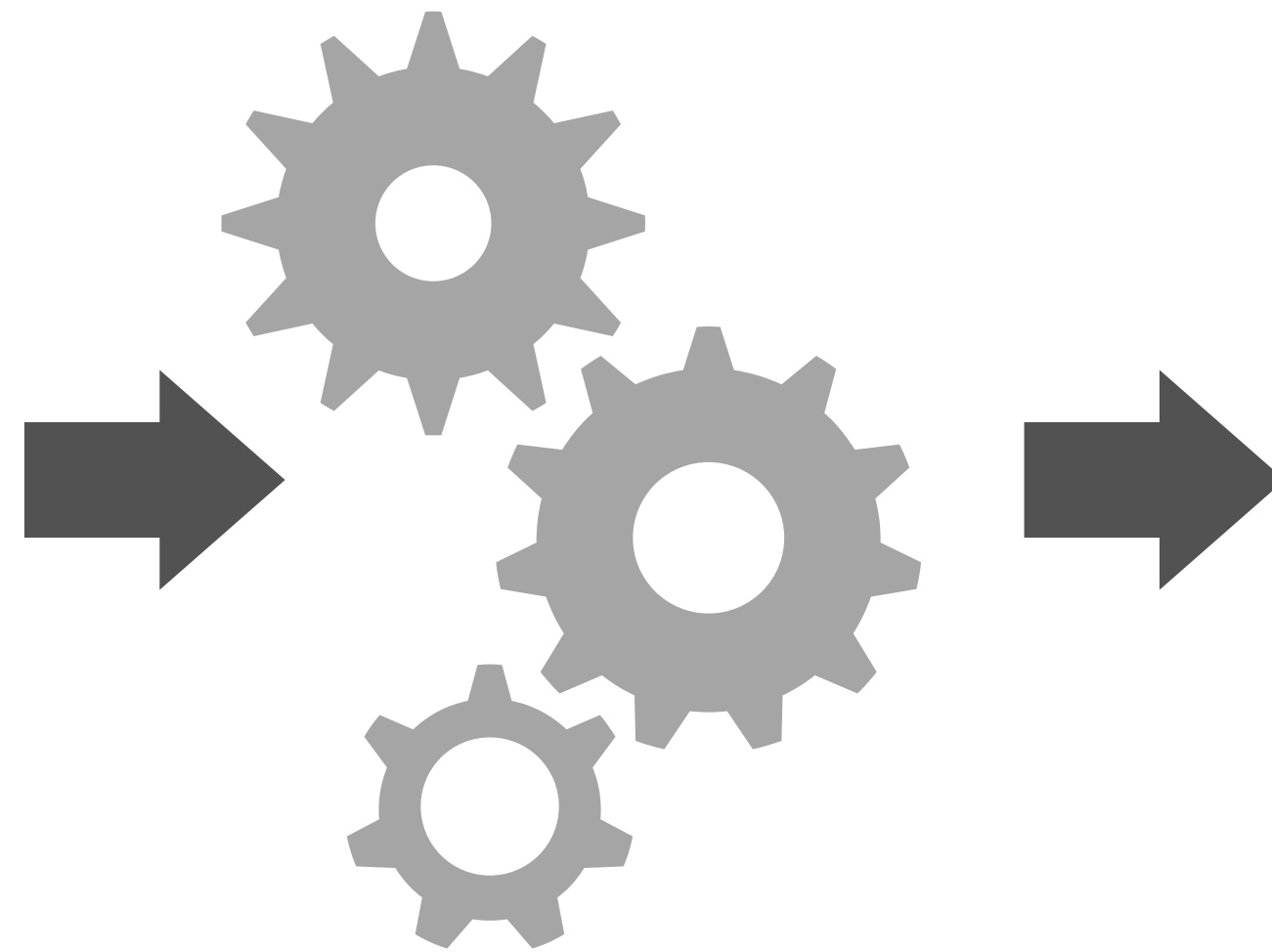
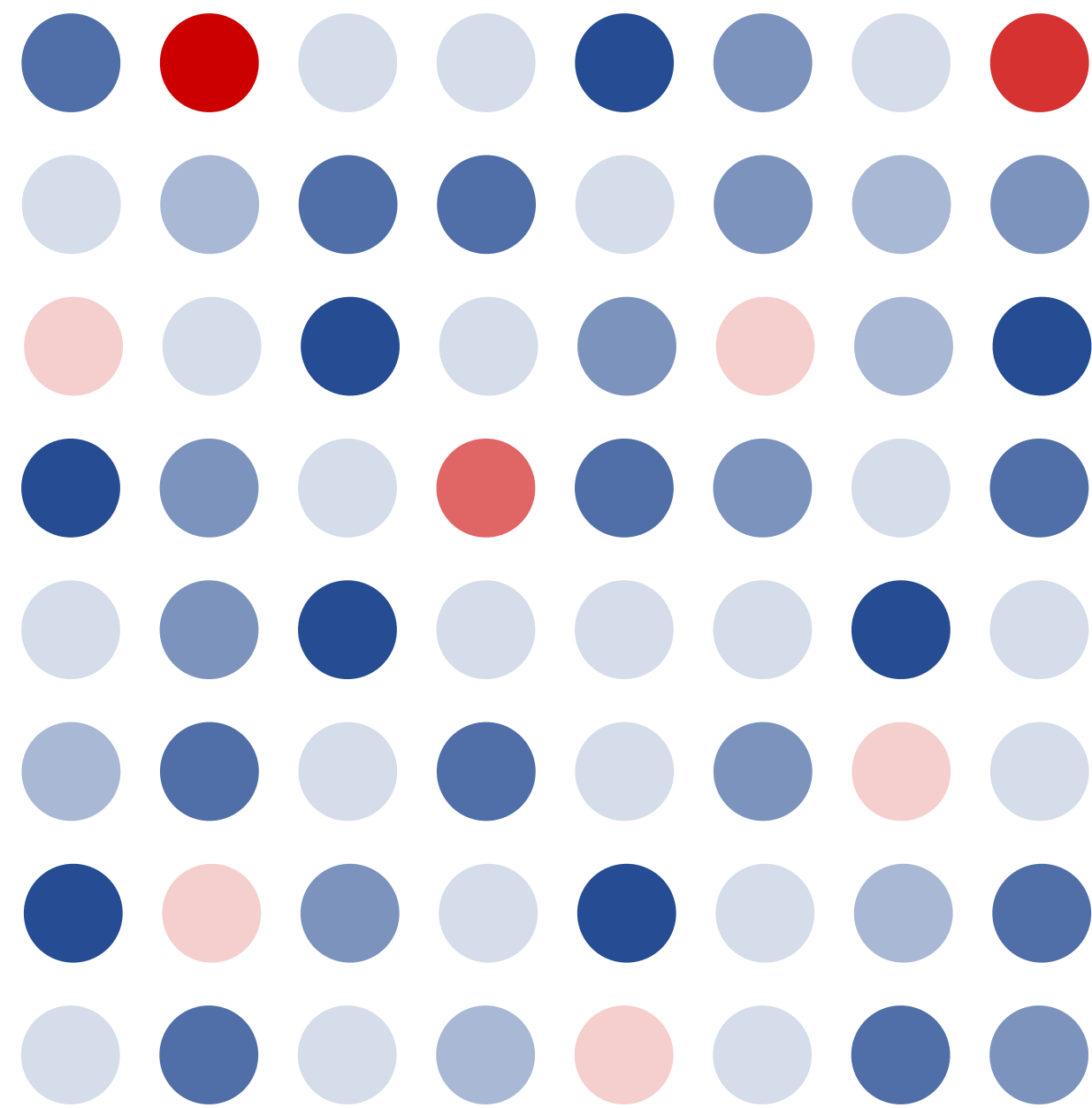




@willb #bbuzz



@willb #bbuzz





Scale

Sophie Watson

**Dealing with pain points of
recommenders in the real world**

06/17/2019 - 14:00 to 14:40

Palais Atelier

long talk (40 min)

X = 0.02381

approximate

reciprocal of 42

$x = 0.02381$

x : Threshold = 0.02381



Stream

Erik Erlandson

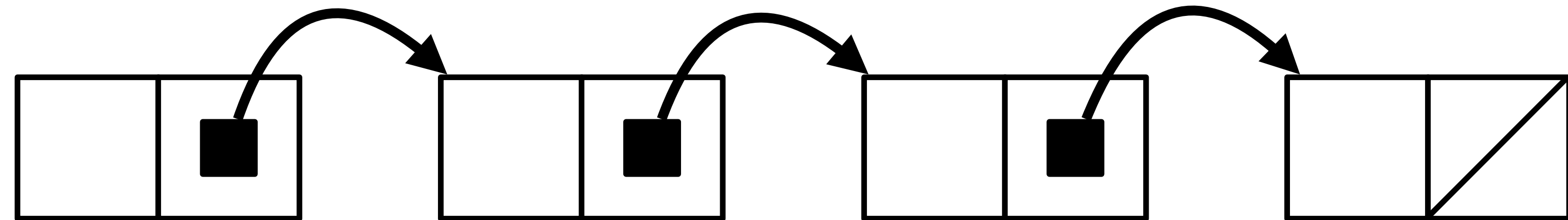
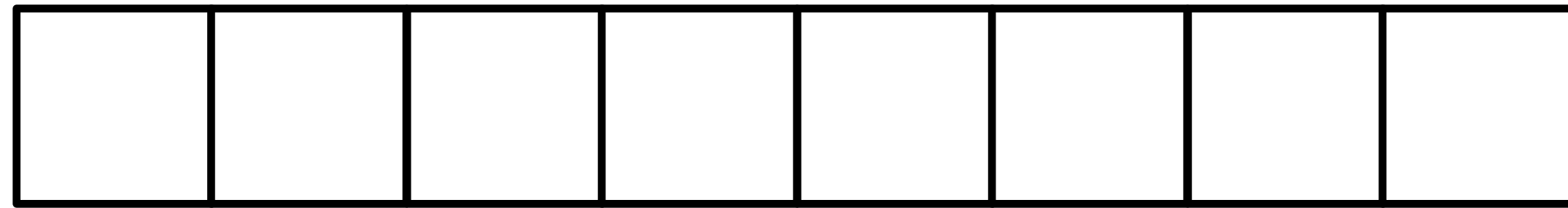
Miles or Kilometers? Why Your Data Schema Should Include Units

06/17/2019 - 11:50 to 12:10

Frannz Salon

short talk (20 min)

What do you mean by “List”?



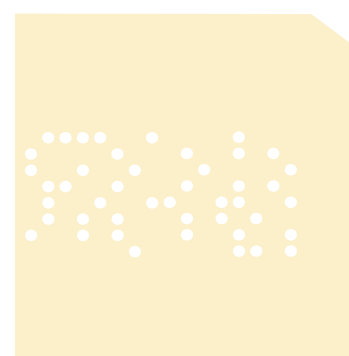
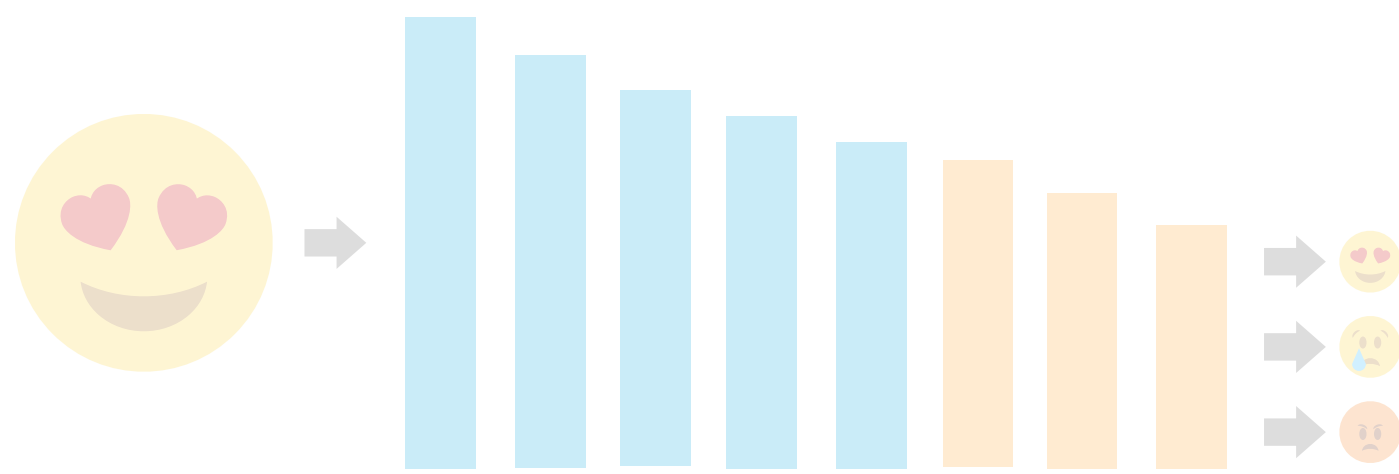
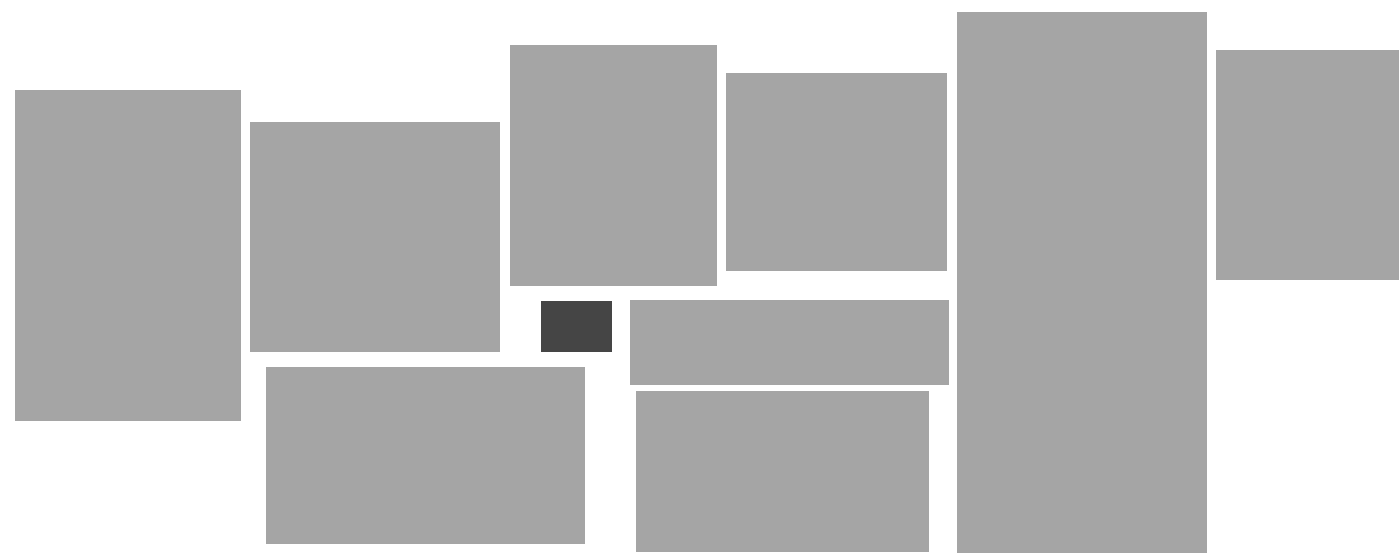
x : Threshold = 0.02381

x : ~~Threshold~~ = 0.02381

x : Float = 0.02381

What have we discussed?

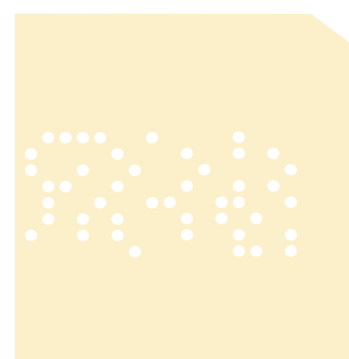
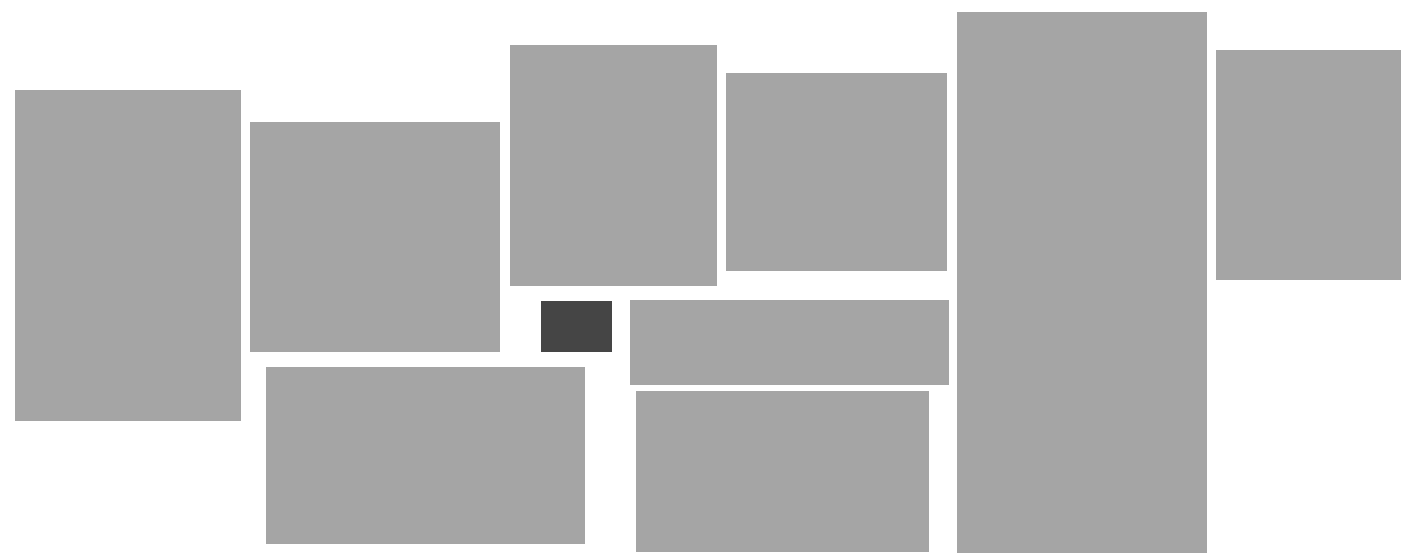




```
def dot(xs, ys):  
    return sum([x * y for  
                x, y in zip(xs, ys)])
```



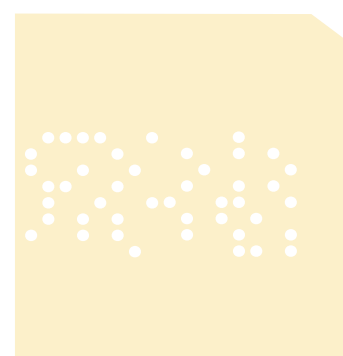
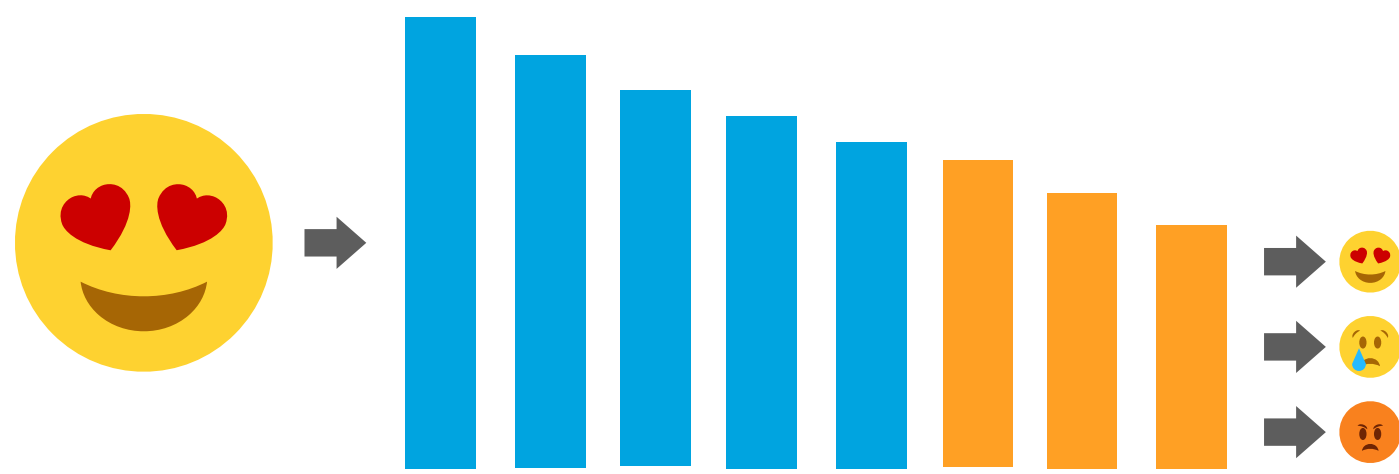
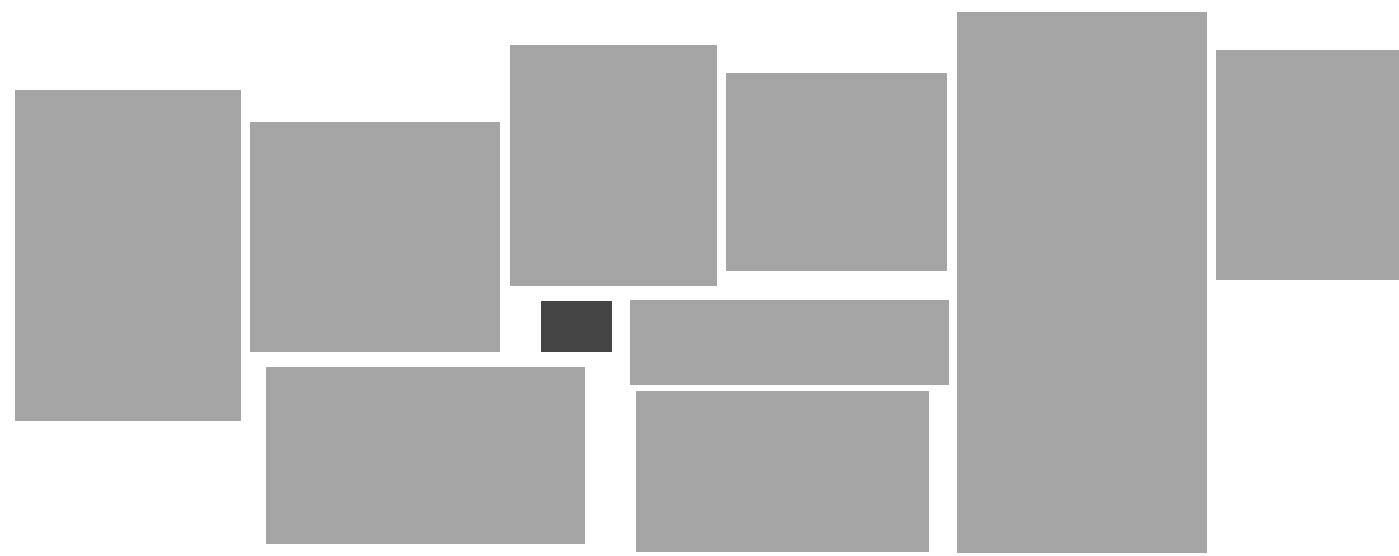
Sculley et al., “Hidden Technical Debt in Machine Learning Systems.” *NIPS 2015*



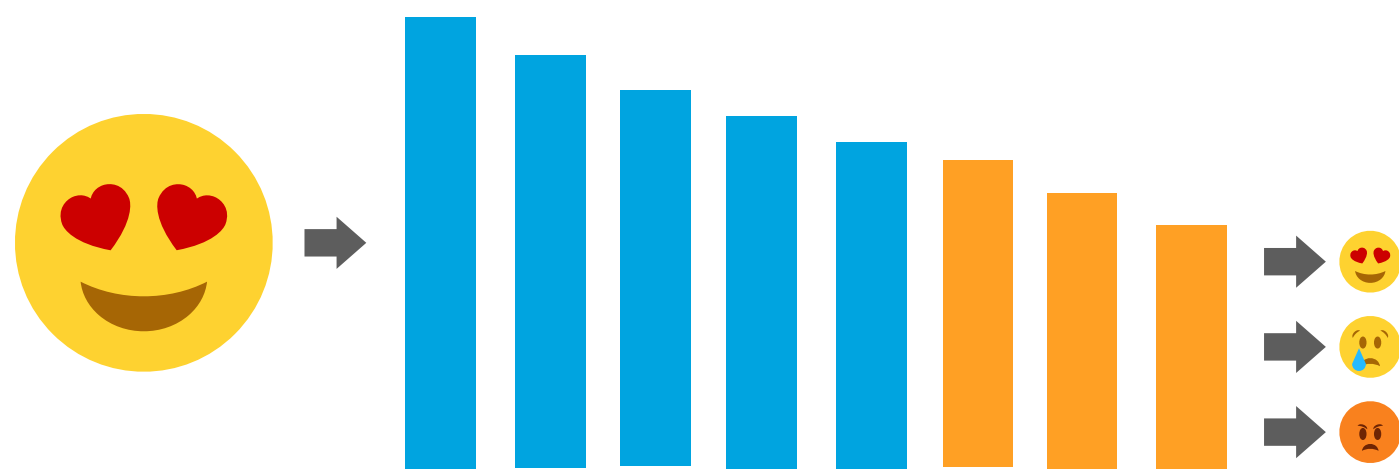
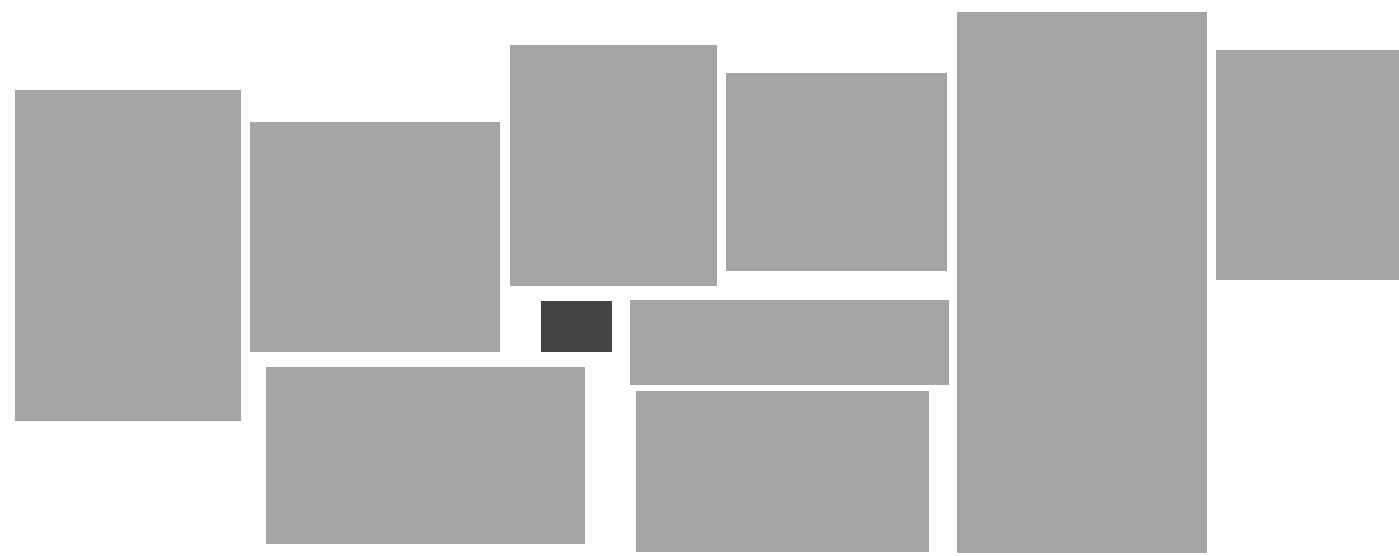
```
def dot(xs, ys):  
    return sum([x * y for  
                x, y in zip(xs, ys)])
```



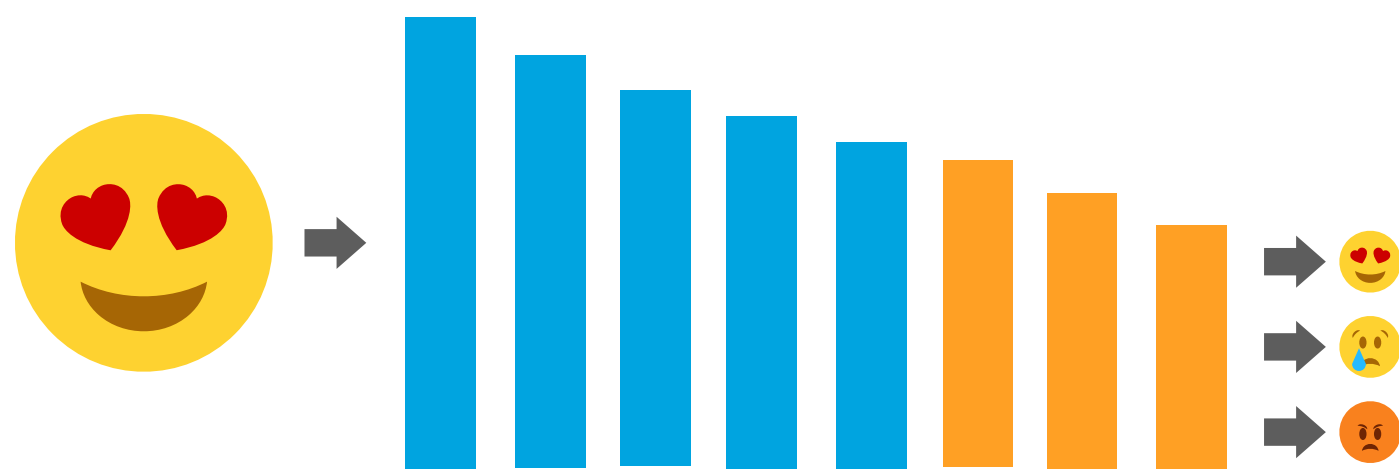
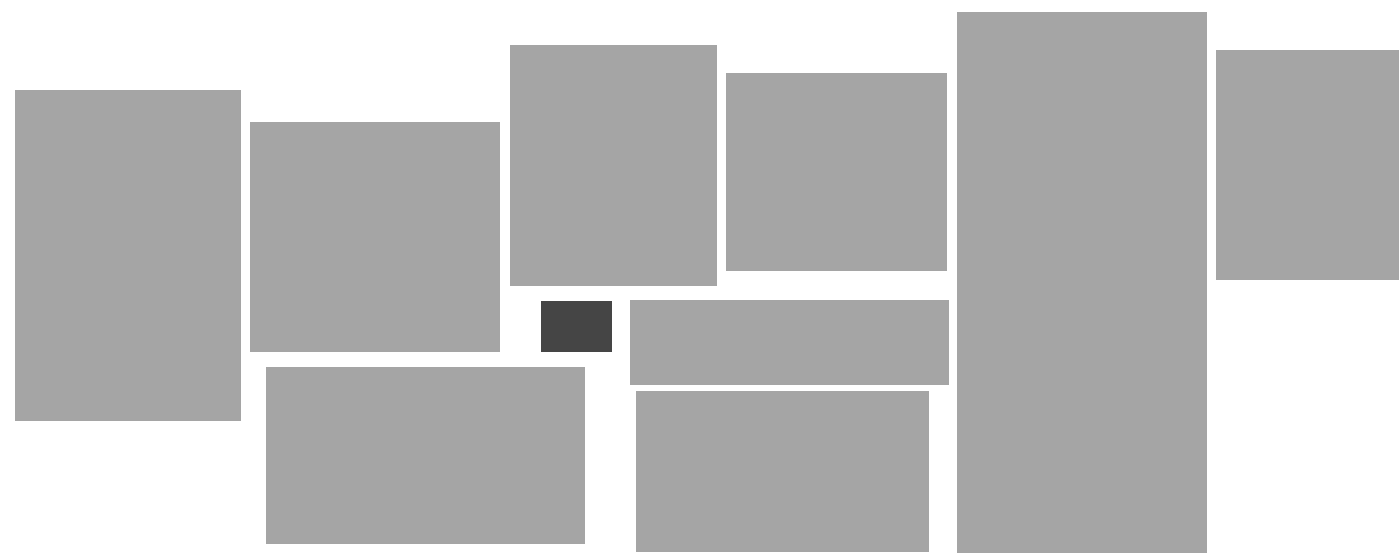
Szegedy et al., “Intriguing properties of neural networks.” In *ICLR '14*.



```
def dot(xs, ys):  
    return sum([x * y for  
                x, y in zip(xs, ys)])
```



```
def dot(xs, ys):  
    return sum([x * y for  
                x, y in zip(xs, ys)])
```



```
def dot(xs, ys):  
    return sum([x * y for  
                x, y in zip(xs, ys)])
```



Search

Szilard Pafka

Better than Deep Learning: Gradient Boosting Machines

06/17/2019 - 15:20 to 16:00

Palais Atelier



Scale

Sophie Watson

Dealing with pain points of recommenders in the real world

06/17/2019 - 14:00 to 14:40

Palais Atelier



Stream

Erik Erlandson

Miles or Kilometers? Why Your Data Schema Should Include Units

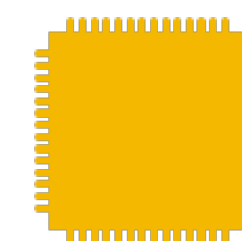
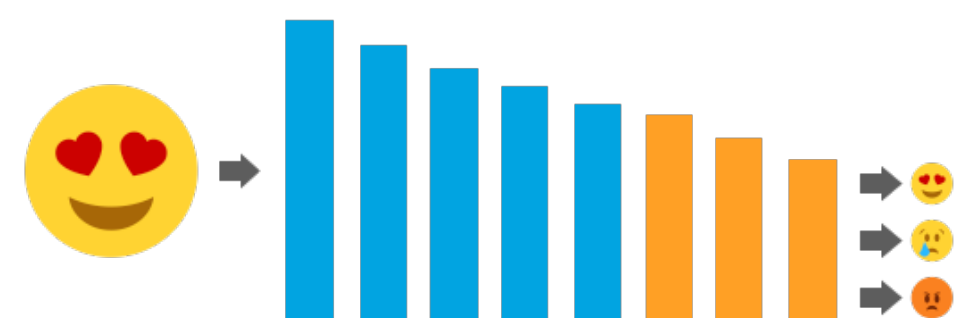
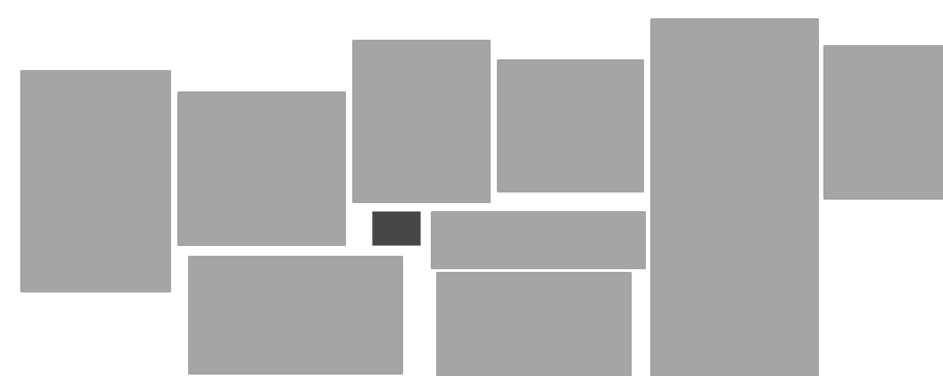
06/17/2019 - 11:50 to 12:10

Frannz Salon

THANK YOU

KEEP IN TOUCH:

@willb • willb@redhat.com



```
def dot(xs, ys):  
    return sum([x * y for  
                x, y in zip(xs, ys)])
```