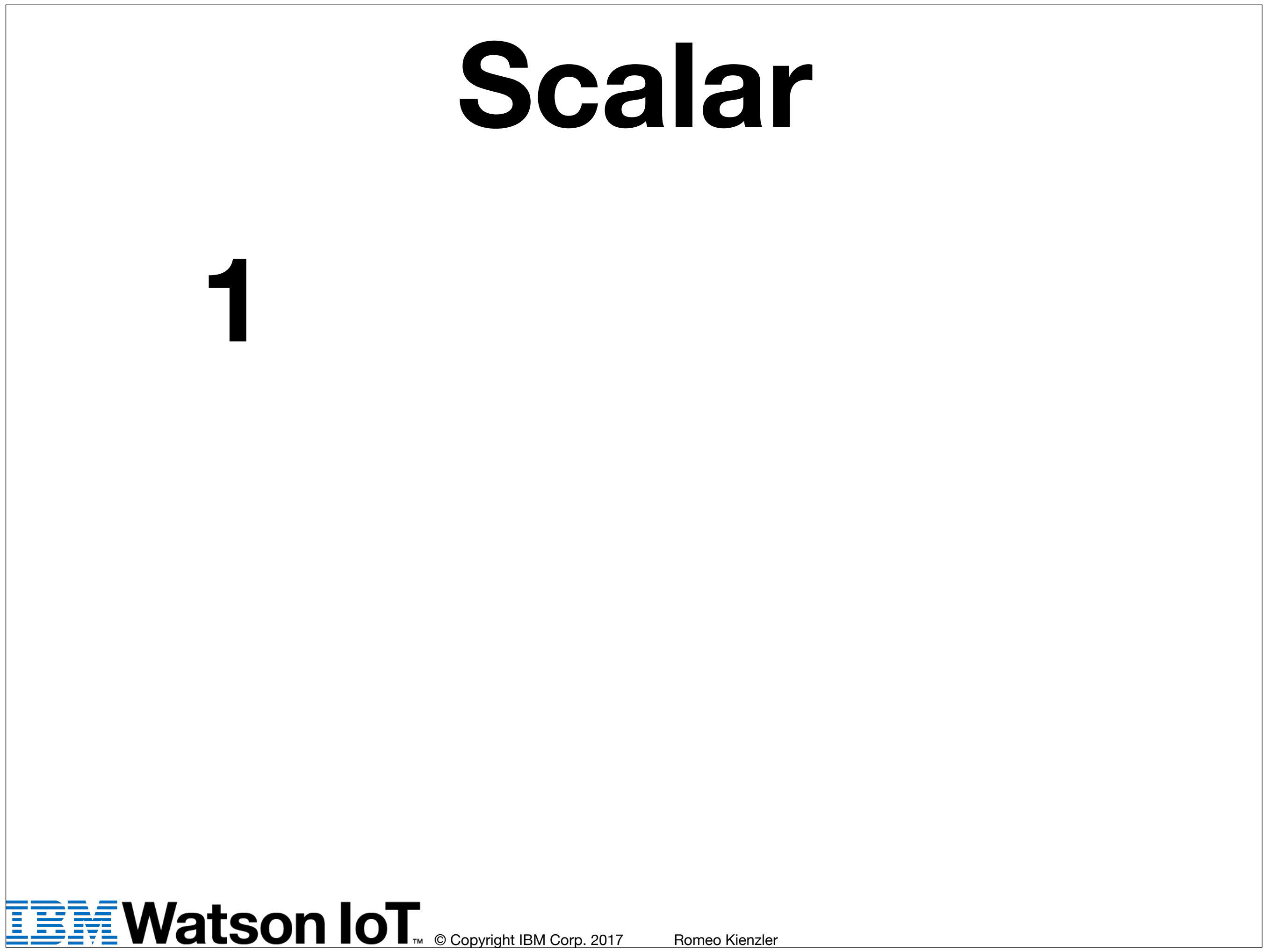
Linear Algebra

Welcome to the lecture on linear algebra

Scalar

So before we start with the math let’s have a look at some important terms. Let’s start with scalar.



Scalar

1

IBM Watson IoT™ Romeo Kienzler

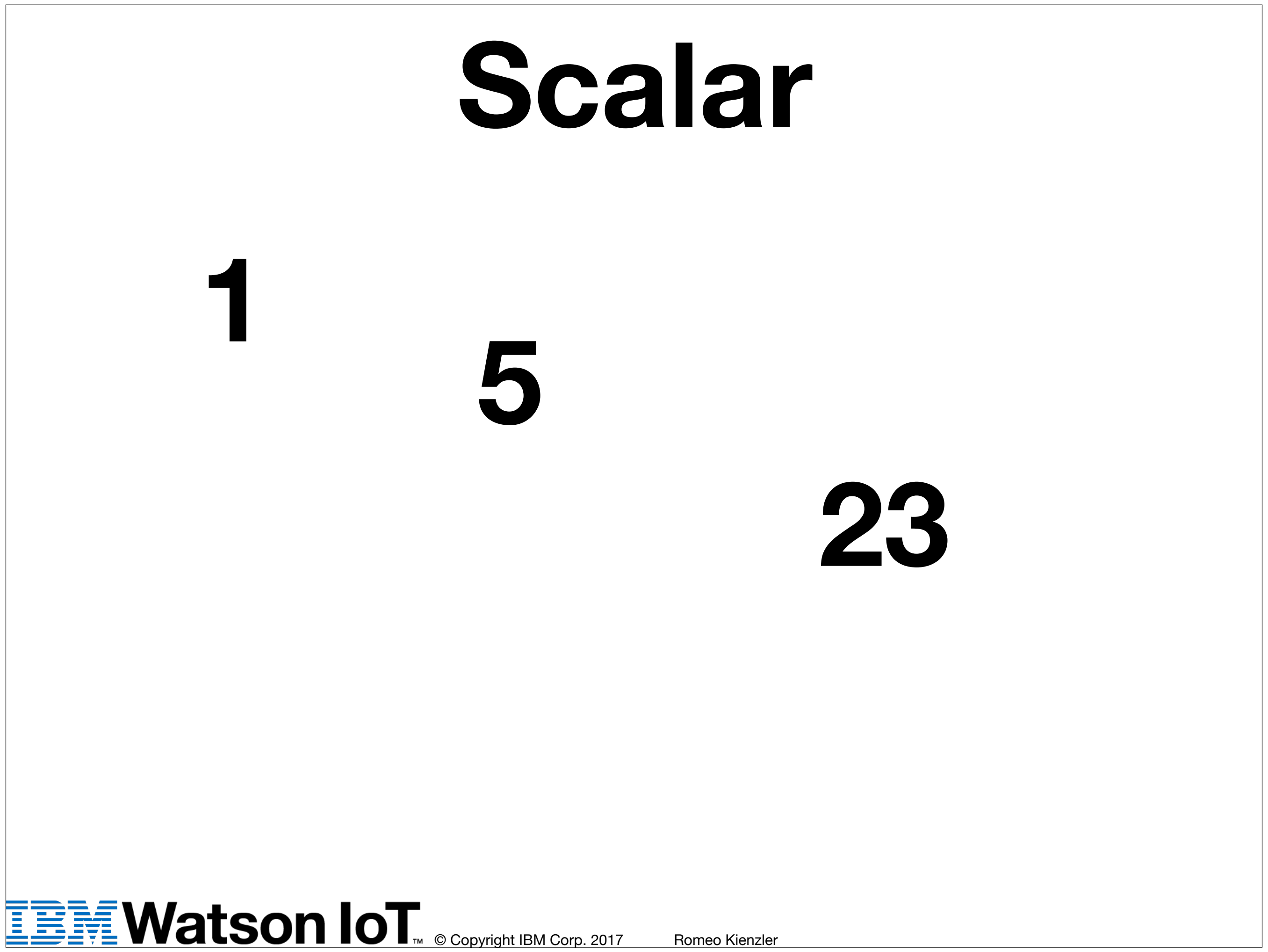
A scalar is basically any number … like one…

Scalar

1

5

..five..



Scalar

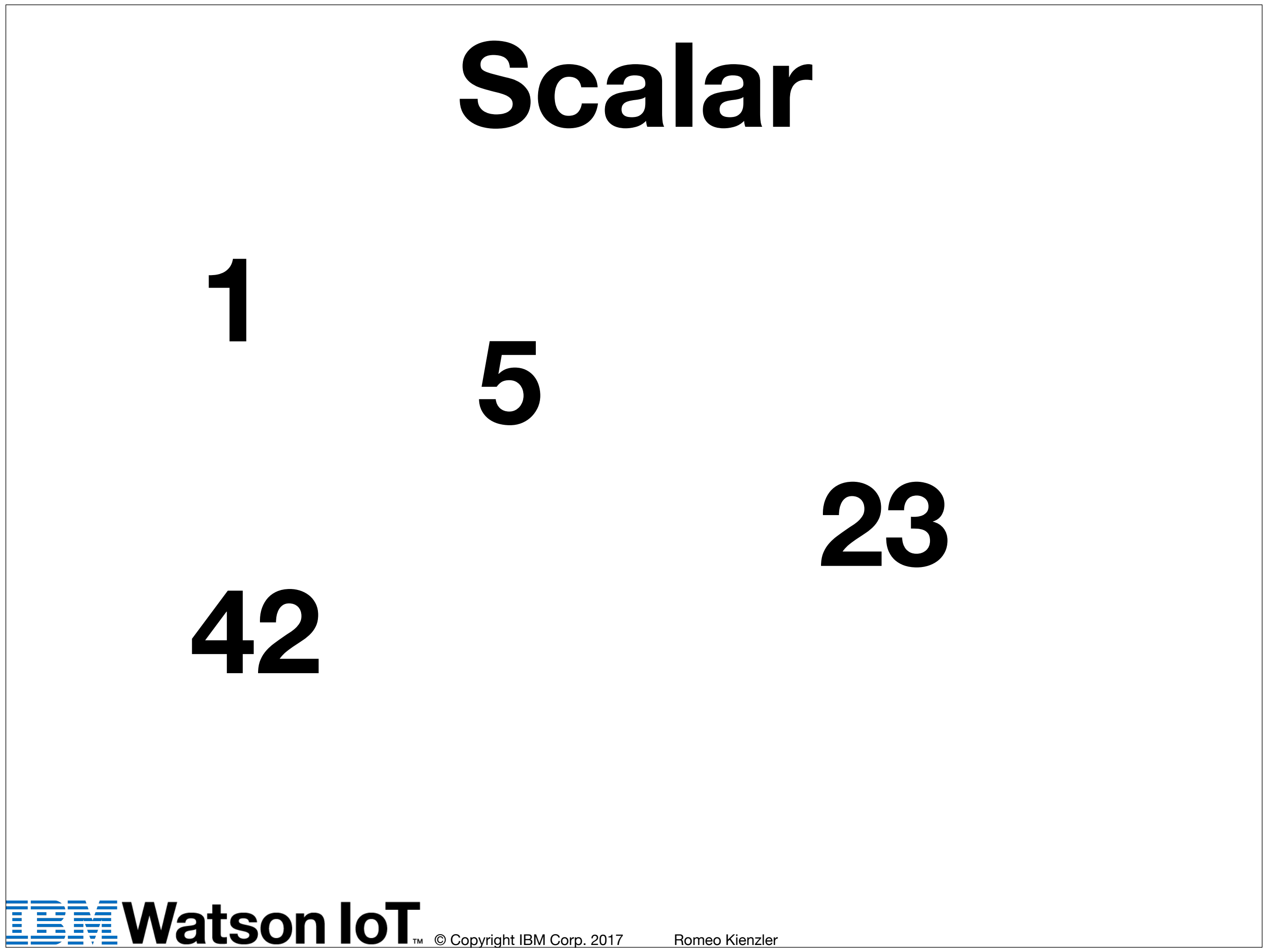
1

5

23

IBM Watson IoT™ Romeo Kienzler

..twenty-three…



Scalar

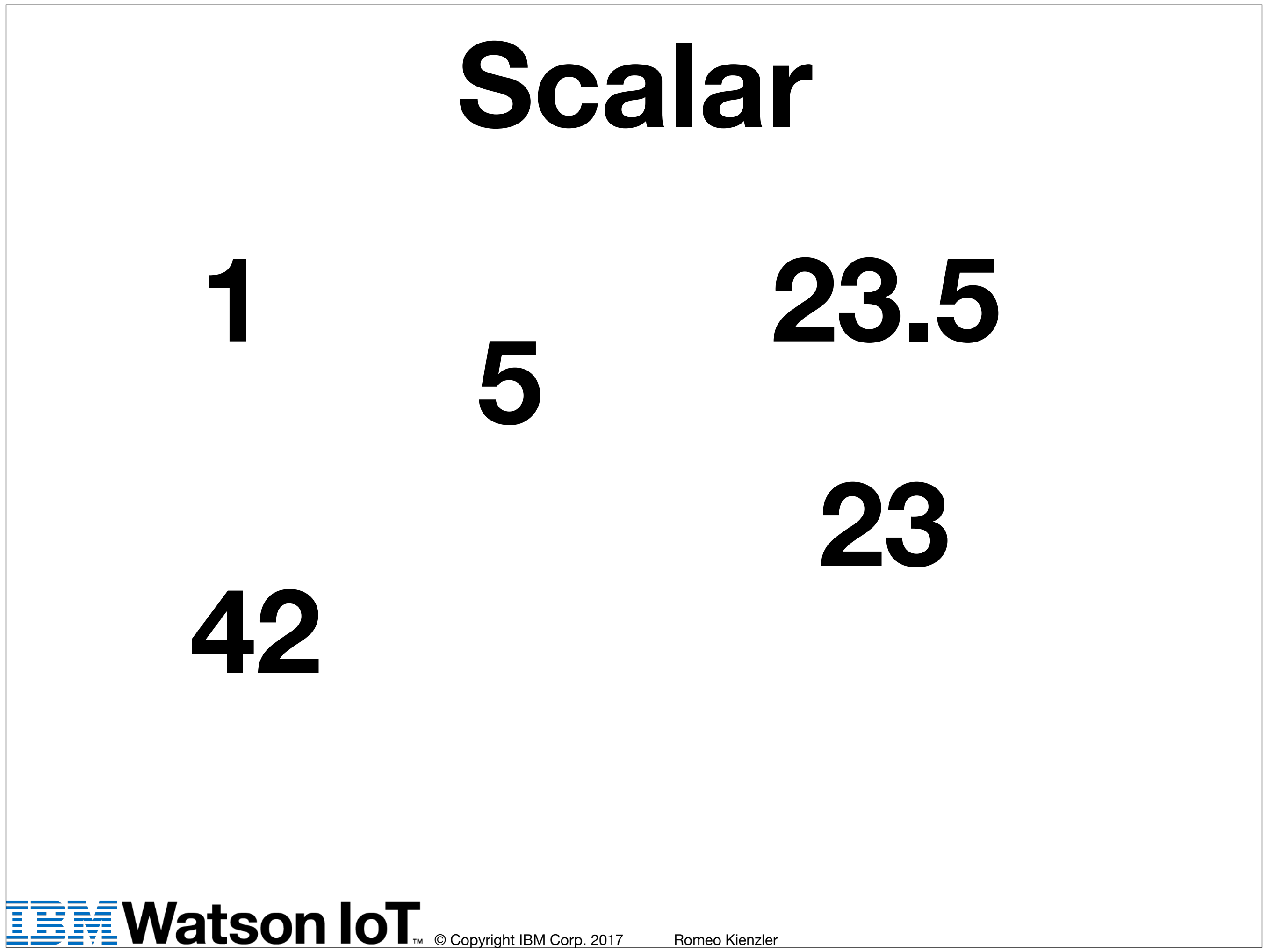
1

5

23

42

or fourth two….



Scalar

1

23.5

5

23

42

but we are not limited to integers here…also a real number like twenty three dot five is a scalar…

Vector

$(0,1,1,2)$

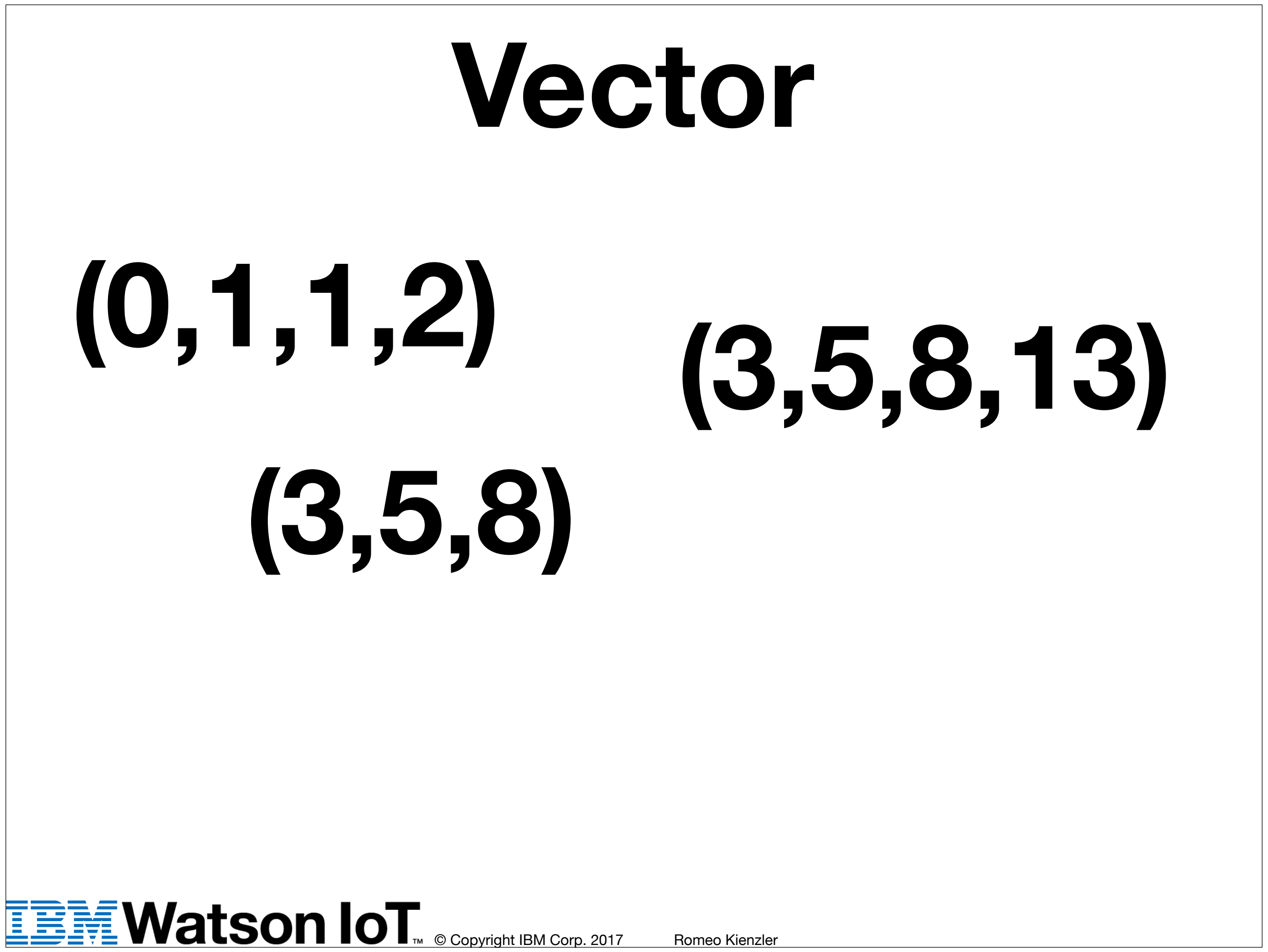
So if you group a bunch of scalars together you'll get a vector…so this is a vector of length four…

Vector

(0,1,1,2)

(3,5,8)

…and this is one of length three…



Vector

(0,1,1,2)

(3,5,8,13)

(3,5,8)

IBM Watson IoT™ Romeo Kienzler

and this one of length five…

Tuple

So a vector can be easily confused with a tuple…

Tuple

(0,1,1,2)

so this guy here can either be a vector or a tuple…

Tuple

(0,1,1,2)

(0,1,2.1,2.3)

but this one is a tuple…can you see the difference? so in a vector, each element needs to have the same type, whereas in a tuple types can be mixed…

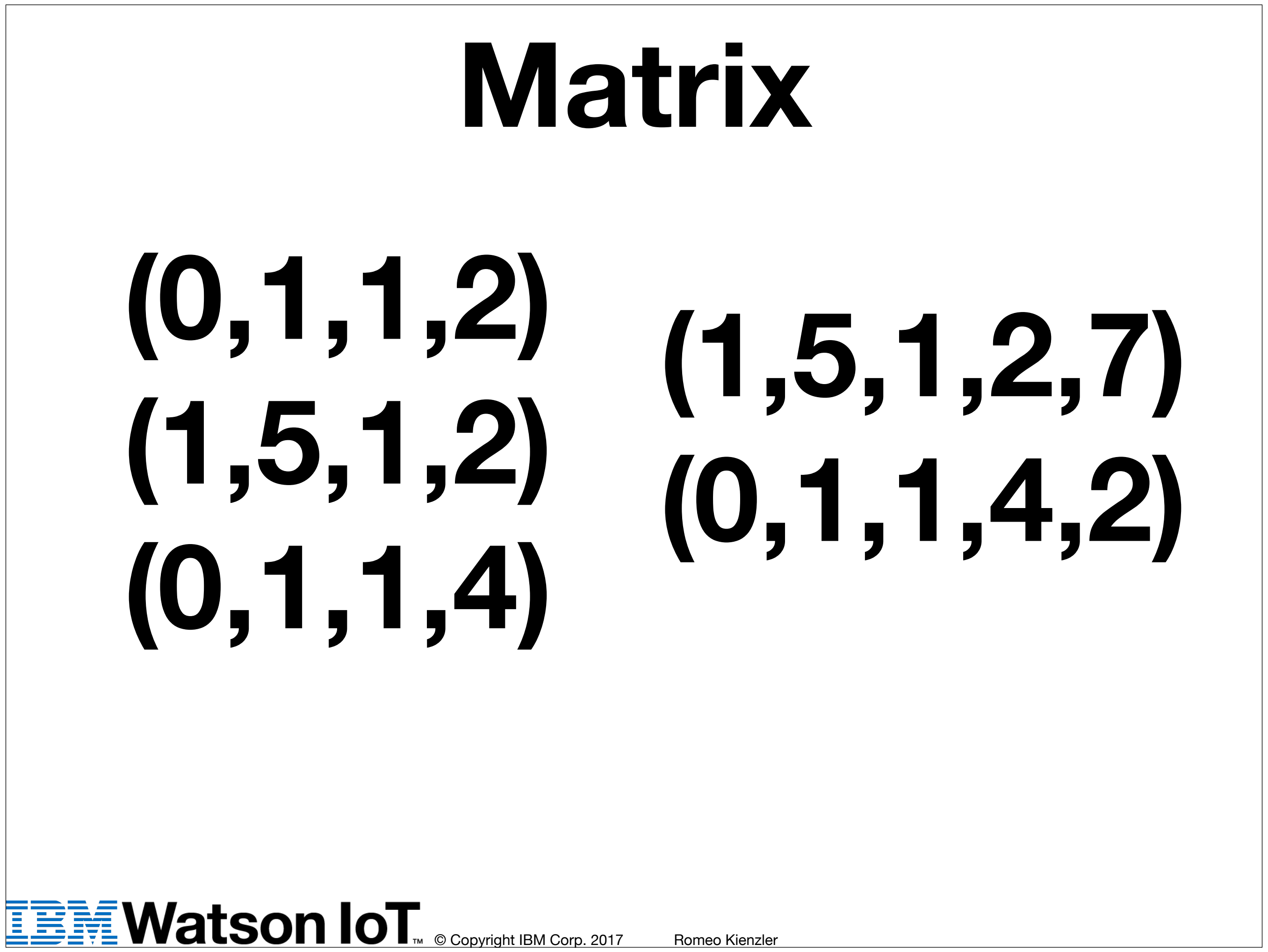
Matrix

A matrix is the big brother of a vector

Matrix

(0,1,1,2)
(1,5,1,2)
(0,1,1,4)

it is basically a list of equal sized vectors



Matrix

(0,1,1,2)
(1,5,1,2)
(0,1,1,4)

(1,5,1,2,7)
(0,1,1,4,2)

note that the number of rows and columns can be different, but again, all elements need to have the same type

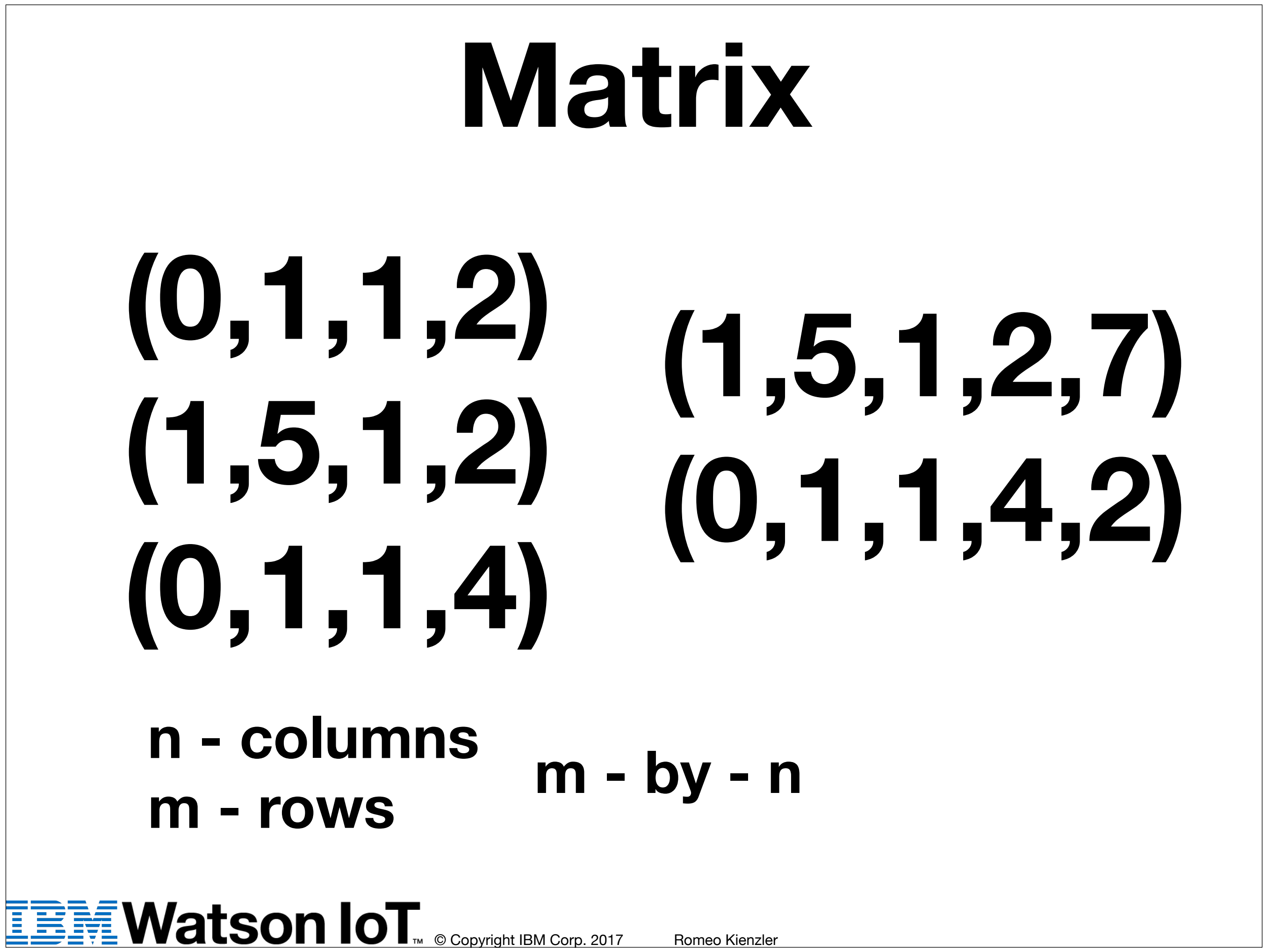
Matrix

(0,1,1,2)
(1,5,1,2)
(0,1,1,4)

(1,5,1,2,7)
(0,1,1,4,2)

n - columns
m - rows

any matrix has n columns and m rows



Matrix

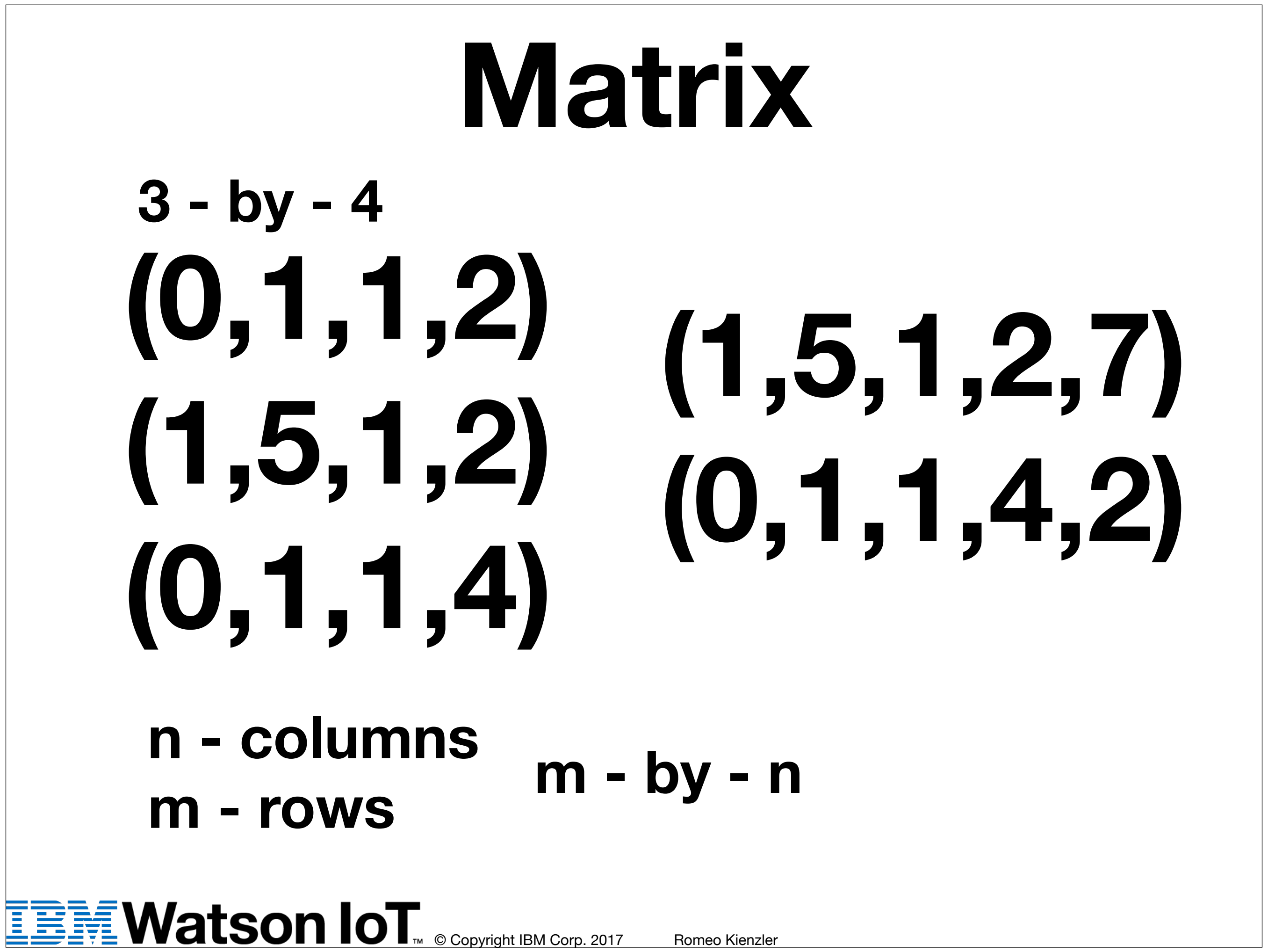
(0,1,1,2)
(1,5,1,2)
(0,1,1,4)

(1,5,1,2,7)
(0,1,1,4,2)

n - columns
m - rows

m - by - n

therefore we call those matrices m - by - n matrix



Matrix

3 - by - 4

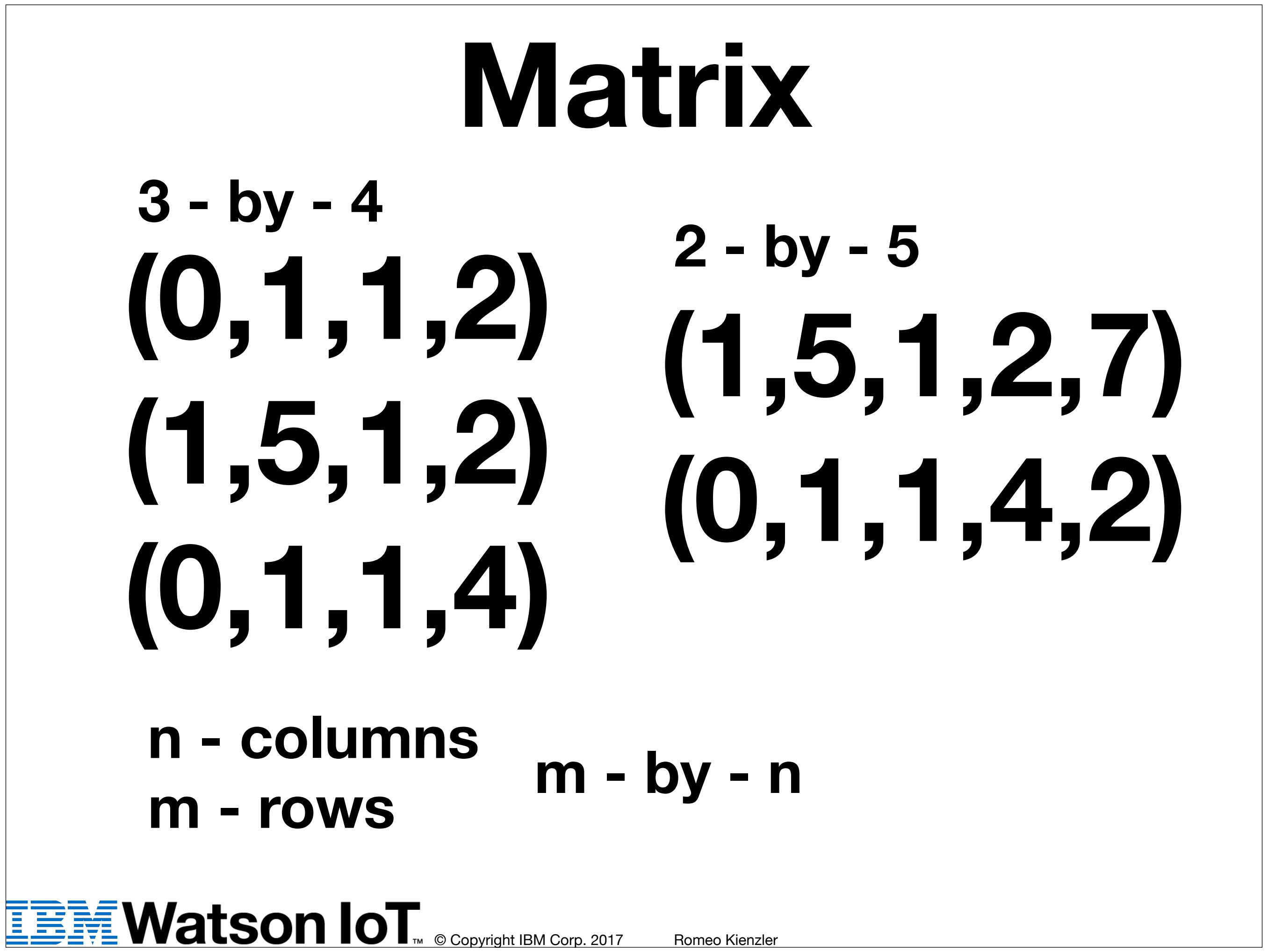
(0,1,1,2)
(1,5,1,2)
(0,1,1,4)

(1,5,1,2,7)
(0,1,1,4,2)

n - columns
m - rows

m - by - n

so the left one is a three by four matrix



Matrix

3 - by - 4

(0,1,1,2)
(1,5,1,2)
(0,1,1,4)

2 - by - 5

(1,5,1,2,7)
(0,1,1,4,2)

n - columns
m - rows

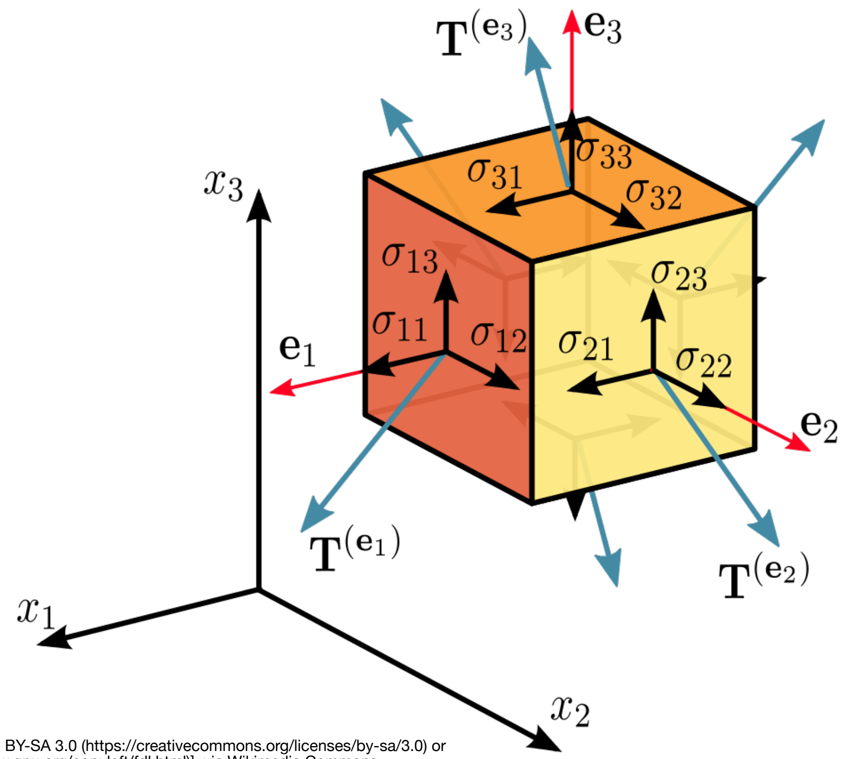
m - by - n

Watson IoT™

Romeo Kienzler

whereas the right one is a two by five matrix

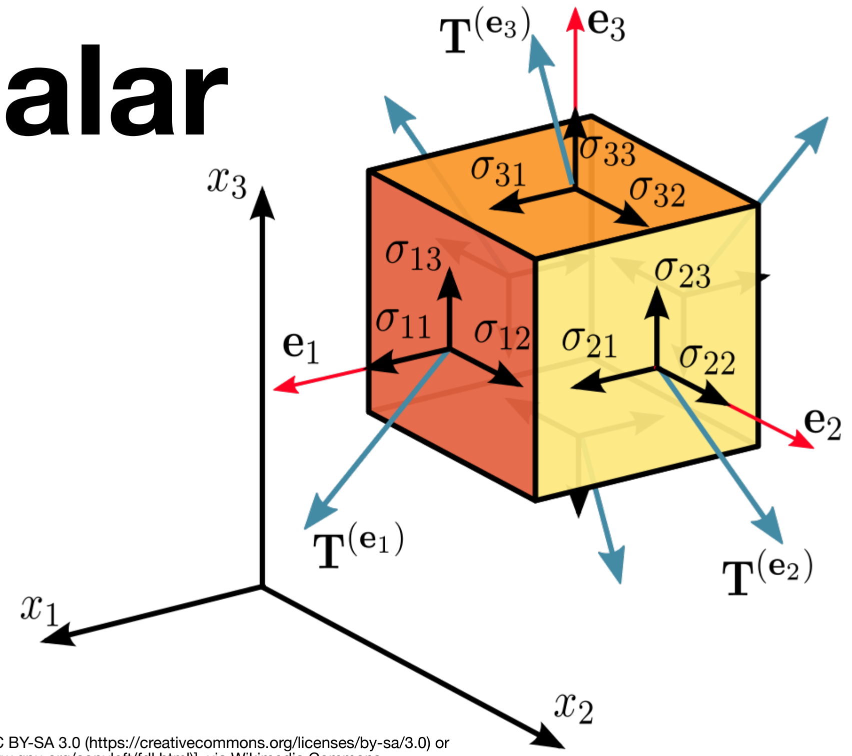
Tensor



finally, the coolest guy is called a tensor. here you see a three dee tensor which is basically nothing else than a matrix with three instead of two dimension. three dee tensors are quite handy for image processing since beside two dimensional pixel information also colour, alpha and focus information can be expressed. tensor is a more general term for specials cases

Tensor

Scalar

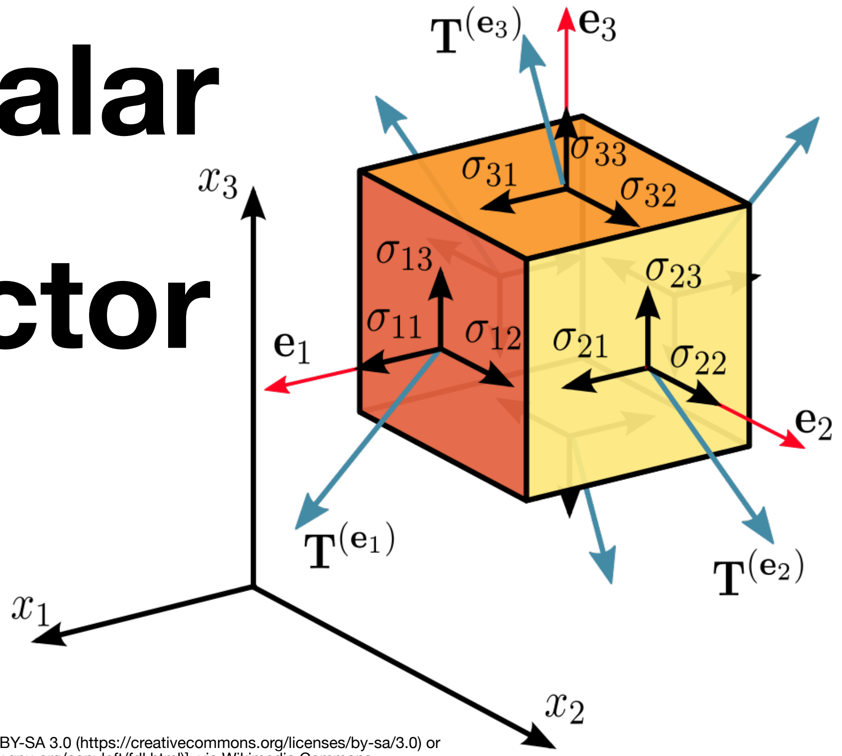


…so for example a zero dimensional tensor is called scalar…

Tensor

Scalar

Vector



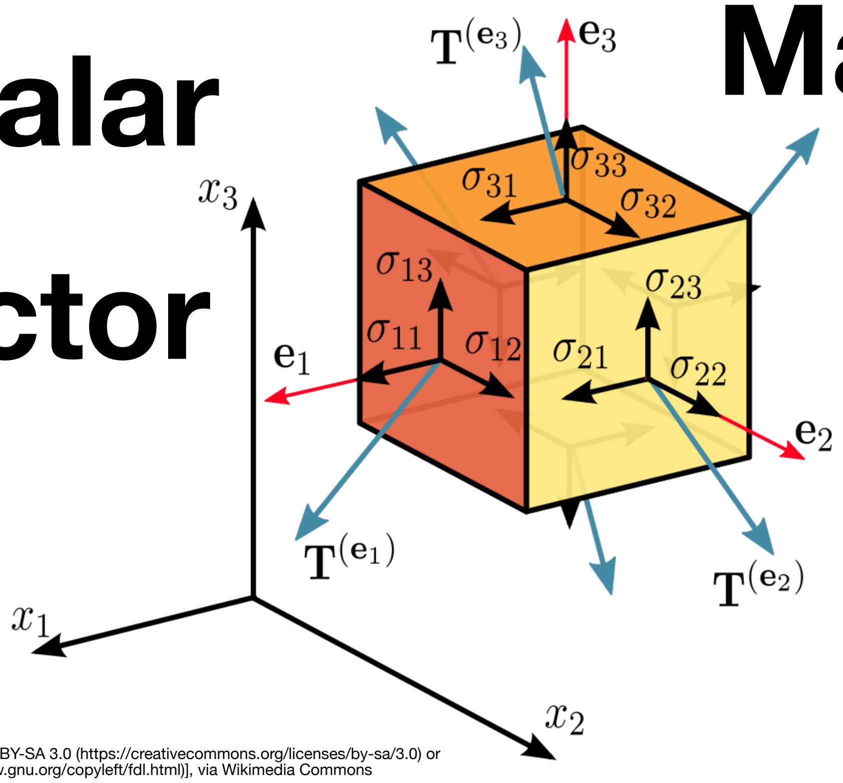
a one dimensional tensor is called vector…

Tensor

Matrix

Scalar

Vector



..and a two dimensional tensor is called matrix…so you can really show off in front of your colleagues by knowing all those terms :) so let's start with some math…

Multiplication

the most important mathematical operation on tensors used in this course is multiplication

Scalar Multiplication

$$2 * 3 = 6$$

we can multiply two scalars

a3_m1_v1_ex1_1

So let start with two integers three and two. If we multiply them we get six as expected.

Scalar Multiplication

$$\begin{bmatrix} 0 \\ 1 \\ 3 \\ 6 \end{bmatrix} * \begin{bmatrix} 2 \\ 7 \\ 13 \\ 20 \end{bmatrix}$$

but we can also multiply two vectors. note that the column wise notation is the most common in math. this multiplication is called dot product or scalar product because it returns a scalar. we won't cover the cross product or vector product here.

Vector Multiplication

$$\begin{bmatrix}0\\1\\3\\6\end{bmatrix} * \begin{bmatrix}2\\7\\13\\20\end{bmatrix} = 0 * 2$$

so the vector product simply takes the first element of the first vector and multiplies it with the first element of the second vector

Vector Multiplication

$$\begin{bmatrix} 0 \\ 1 \\ 3 \\ 6 \end{bmatrix} * \begin{bmatrix} 2 \\ 7 \\ 13 \\ 20 \end{bmatrix} = 0 * 2 \quad 1 * 7$$

then it does the same for the second elements of the two vectors

Vector Multiplication

$$\begin{bmatrix} 0 \\ 1 \\ 3 \\ 6 \end{bmatrix} * \begin{bmatrix} 2 \\ 7 \\ 13 \\ 20 \end{bmatrix} = 0 * 2 \quad 1 * 7 \quad 3 * 13 \quad 6 * 20$$

and finally for the rest

Vector Multiplication

$$\begin{bmatrix}0\\1\\3\\6\end{bmatrix} * \begin{bmatrix}2\\7\\13\\20\end{bmatrix} = 0*2+1*7+3*13+6*20$$

now all these intermediate products are summed up

Vector Multiplication

$$\begin{bmatrix}0\\1\\3\\6\end{bmatrix} * \begin{bmatrix}2\\7\\13\\20\end{bmatrix} = 0*2+1*7+3*13+6*20 = 166$$

and the resulting sum is the solution to the vector dot product

Vector Multiplication

$$\begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_m \end{bmatrix} * \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{bmatrix} = a_1 * x_1 + a_2 * x_2 + a_3 * x_3 + a_4 * x_4$$

more generally speaking, the vector dot product is the linear combination of those two vectors. and this notation is very handy in neural networks which we will see later because often one vector represents the data and one vectors the parameters to the neural network. also using this notation we can make use of accelerators like SIMD or GPUs. Note that both vectors need to be of the same size

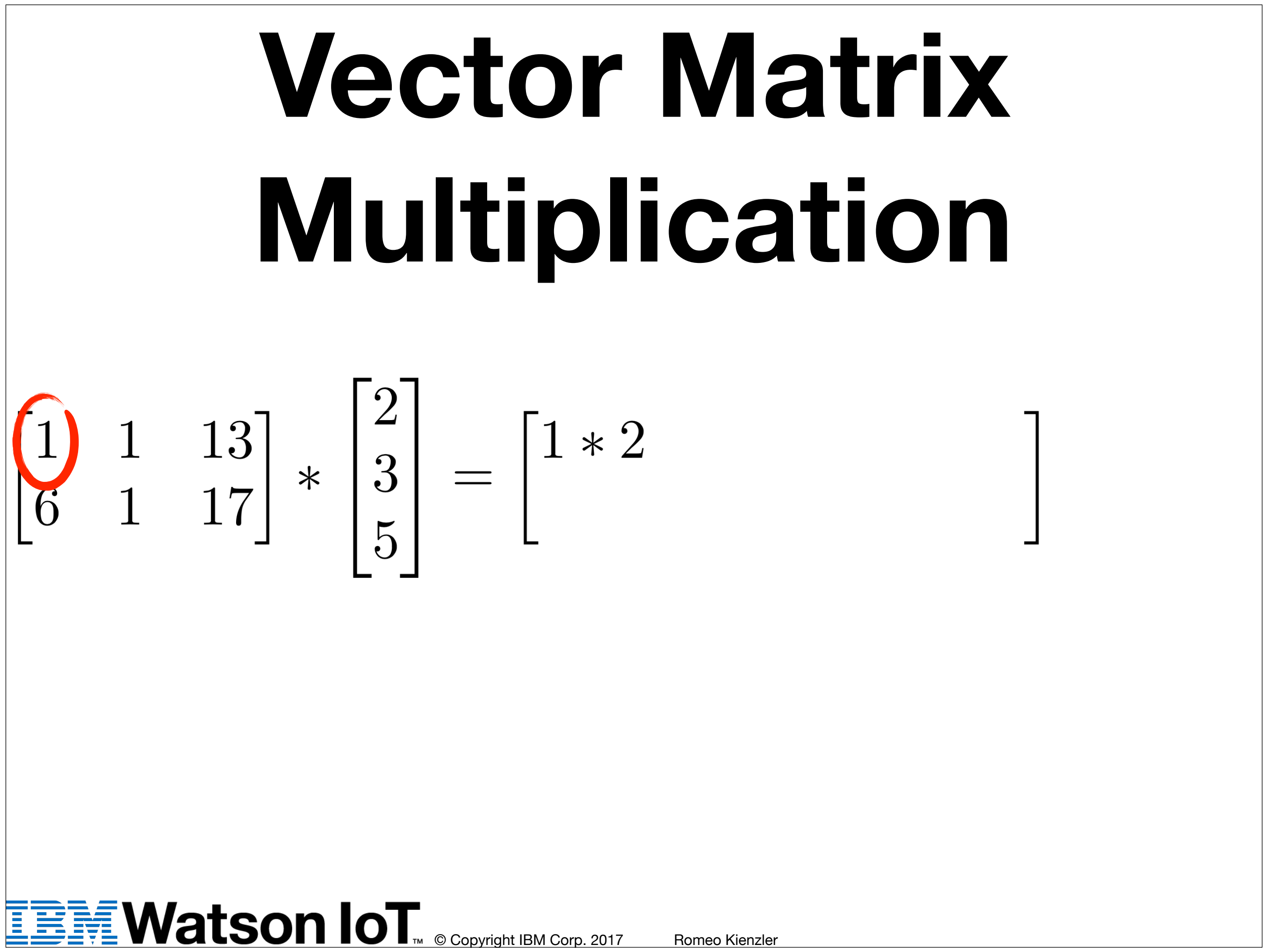
a3_m1_v1_ex1_2

We will use numpy as central library for linear algebra in python. So let's import it. Now we create two vectors a and b. Note that in python those are implemented as one dimensional numpy arrays. Now we can use the dot method of the numpy array object in order to get the dot product of those two vectors.

Vector Matrix Multiplication

$$\begin{bmatrix} 1 & 1 & 13 \\ 6 & 1 & 17 \end{bmatrix} * \begin{bmatrix} 2 \\ 3 \\ 5 \end{bmatrix}$$

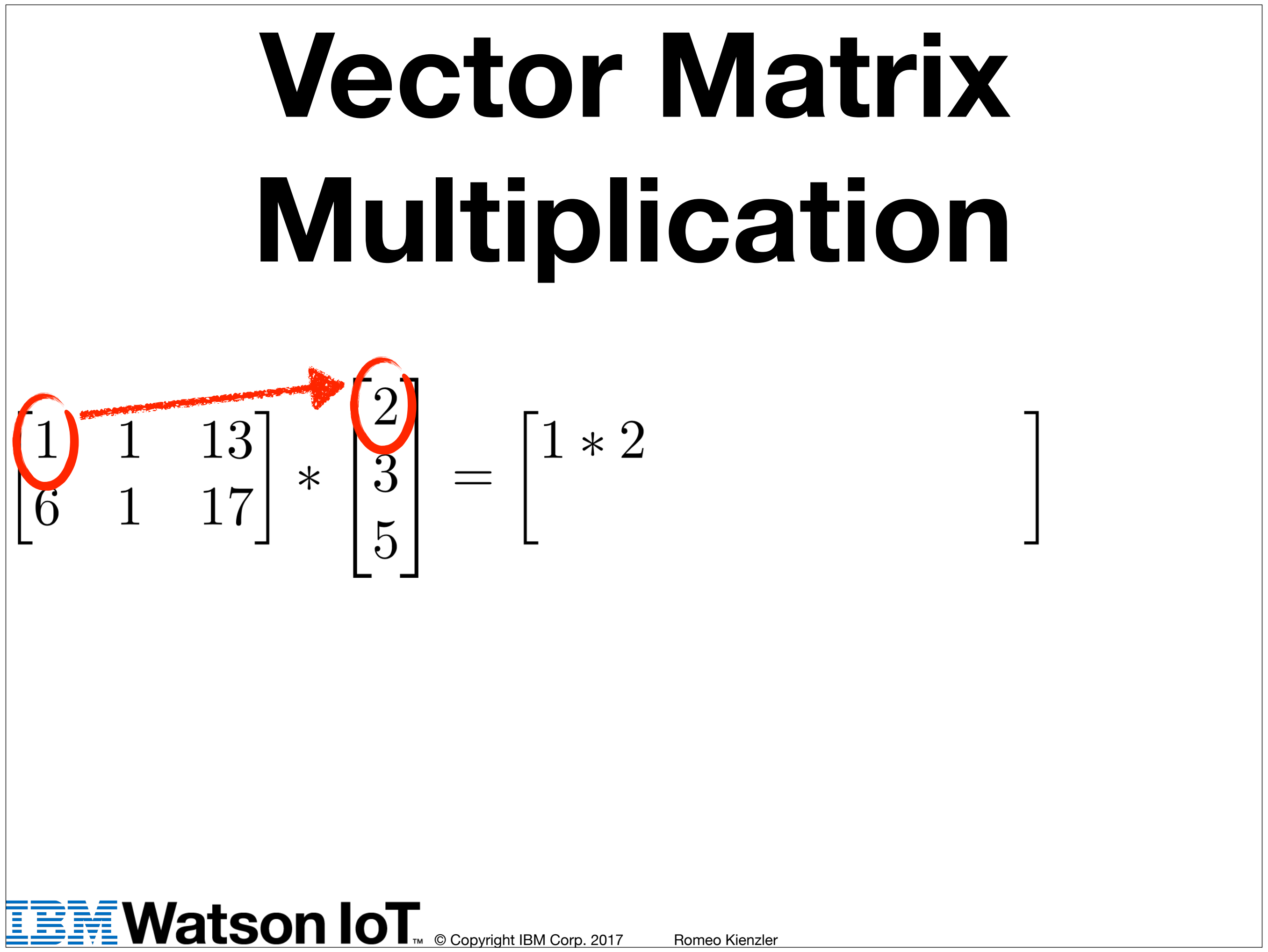
Not only the multiplication of two vectors is defined, but also the multiplication of a matrix with a vector.



Vector Matrix Multiplication

$$\begin{bmatrix} \textcircled{1} & 1 & 13 \\ 6 & 1 & 17 \end{bmatrix} * \begin{bmatrix} 2 \\ 3 \\ 5 \end{bmatrix} = \begin{bmatrix} 1 * 2 & \qquad\qquad\qquad \\ & \end{bmatrix}$$

So just take the element of the first row and column of the matrix..



Vector Matrix Multiplication

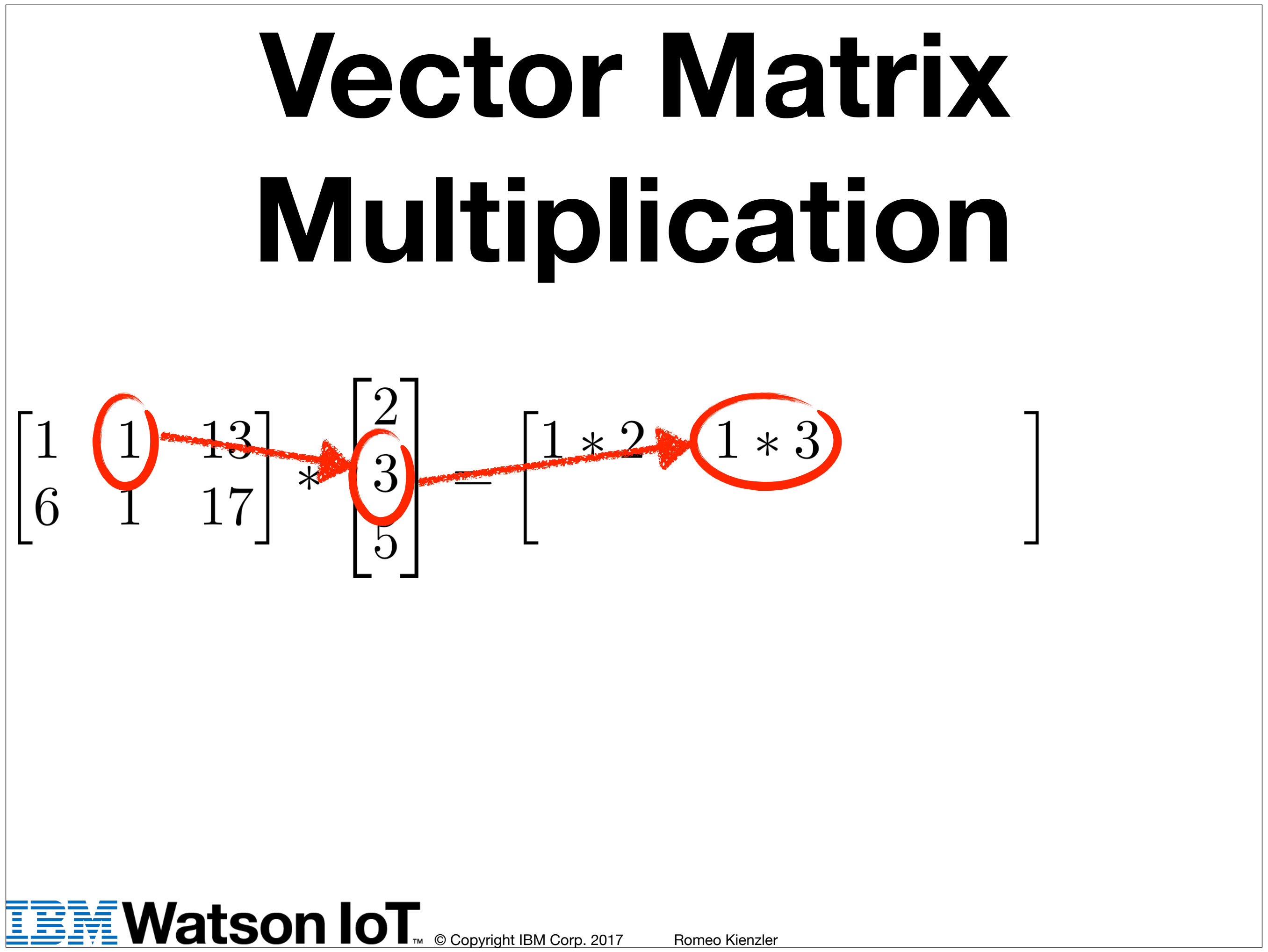
$$\begin{bmatrix} 1 & 1 & 13 \\ 6 & 1 & 17 \end{bmatrix} * \begin{bmatrix} 2 \\ 3 \\ 5 \end{bmatrix} = \begin{bmatrix} 1 * 2 & & \\ & & \end{bmatrix}$$

…and multiply it with the first element of the vector…

Vector Matrix Multiplication

$$\begin{bmatrix} 1 & 1 & 13 \\ 6 & 1 & 17 \end{bmatrix} * \begin{bmatrix} 2 \\ 3 \\ 5 \end{bmatrix} = \begin{bmatrix} 1 * 2 \\ \\ \end{bmatrix}$$

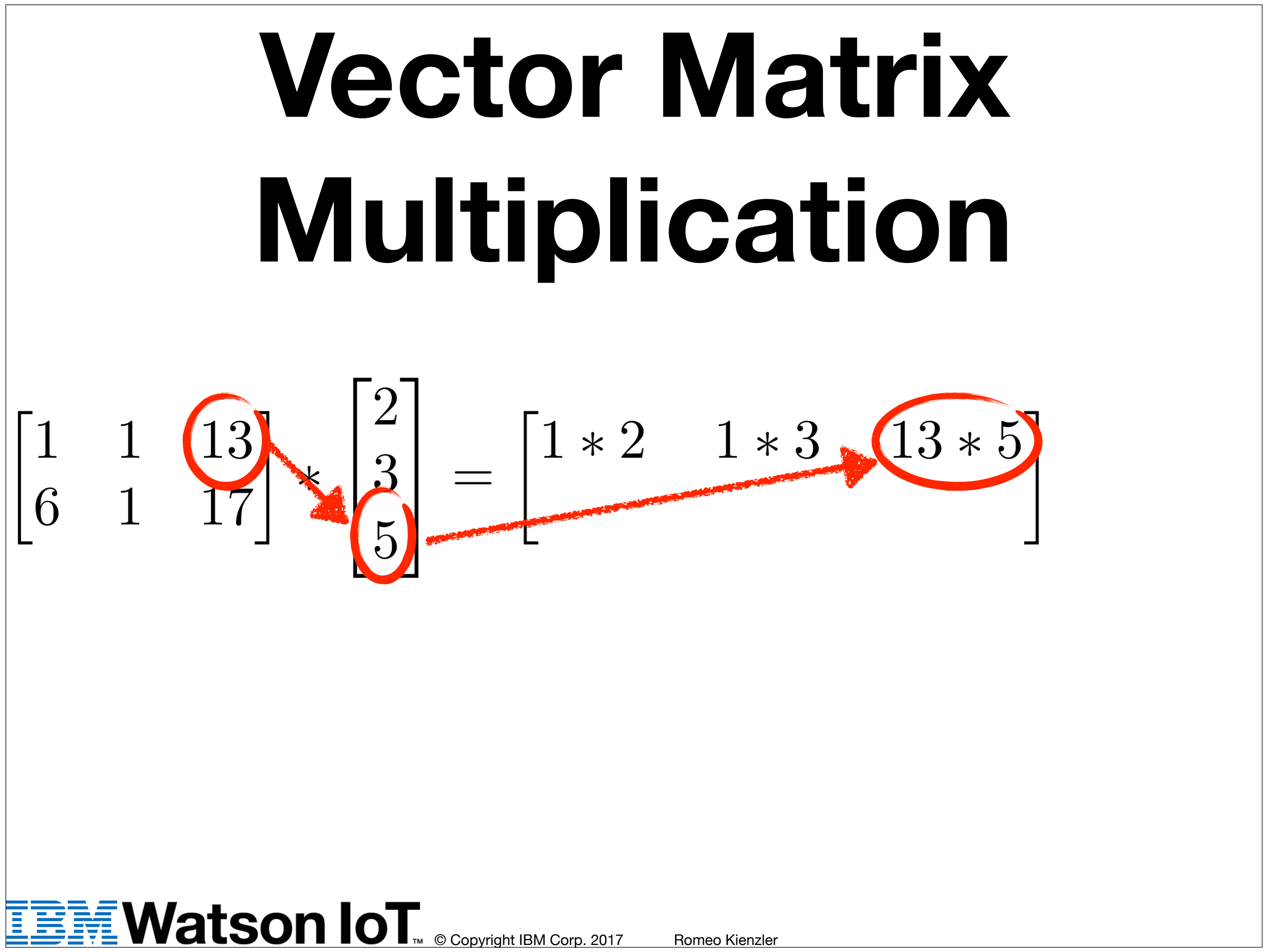
…and put it to the first element of the resulting vector



Vector Matrix Multiplication

$$\begin{bmatrix} 1 & \boxed{1} & 13 \\ 6 & 1 & 17 \end{bmatrix} * \begin{bmatrix} 2 \\ \boxed{3} \\ 5 \end{bmatrix} = \begin{bmatrix} 1*2 & \boxed{1*3} & \\ & & \end{bmatrix}$$

then repeat the steps for the second column first row of the matrix with the second element of the vector



Vector Matrix Multiplication

$$\begin{bmatrix} 1 & 1 & 13 \\ 6 & 1 & 17 \end{bmatrix} * \begin{bmatrix} 2 \\ 3 \\ 5 \end{bmatrix} = \begin{bmatrix} 1*2 & 1*3 & 13*5 \\ & & \end{bmatrix}$$

finally repeat the steps for the third column first row of the matrix with the third element of the vector

Vector Matrix Multiplication

$$\begin{bmatrix} 1 & 1 & 13 \\ 6 & 1 & 17 \end{bmatrix} * \begin{bmatrix} 2 \\ 3 \\ 5 \end{bmatrix} = \begin{bmatrix} 1*2 & 1*3 & 13*5 \\ 6*2 & 1*3 & 17*5 \end{bmatrix}$$

finally we repeat the same steps for the second row of the matrix

Vector Matrix Multiplication

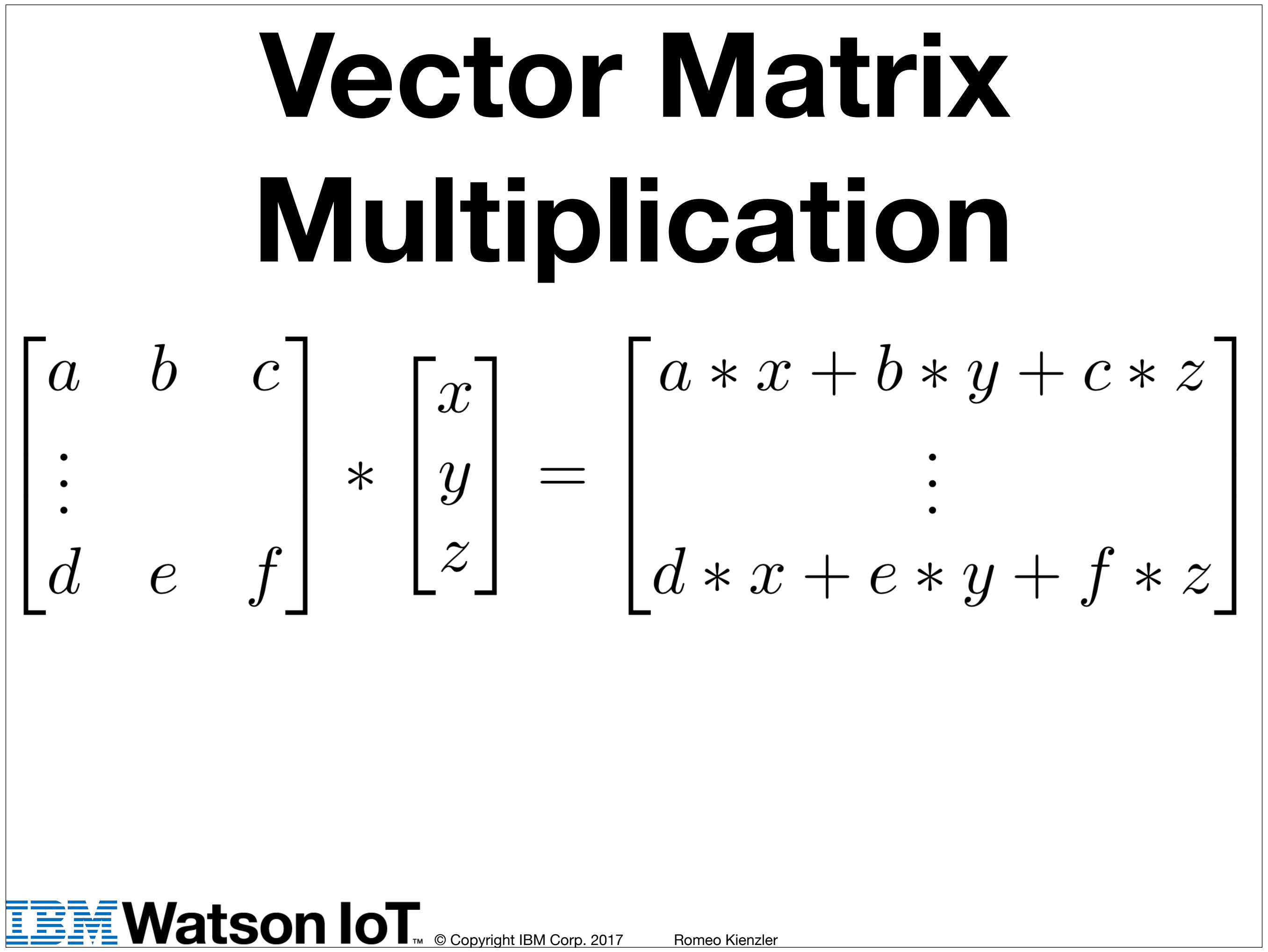
$$\begin{bmatrix} 1 & 1 & 13 \\ 6 & 1 & 17 \end{bmatrix} * \begin{bmatrix} 2 \\ 3 \\ 5 \end{bmatrix} = \begin{bmatrix} 1*2+1*3+13*5 \\ 6*2+1*3+17*5 \end{bmatrix}$$

then we sum up all those components

Vector Matrix Multiplication

$$\begin{bmatrix}1 & 1 & 13\\ 6 & 1 & 17\end{bmatrix} * \begin{bmatrix}2\\ 3\\ 5\end{bmatrix} = \begin{bmatrix}1*2+1*3+13*5\\ 6*2+1*3+17*5\end{bmatrix} = \begin{bmatrix}70\\ 100\end{bmatrix}$$

and the result is a two element vector. so this is nothing else than two vector dot products calculated in one go, where the first vector is encoded in the first row of the matrix and the second vector is encoded in the second row of the matrix. this is also very handy because SIMD and GPU accelerators can speed up things dramatically. also notation becomes far more concise



Vector Matrix Multiplication

$$\begin{bmatrix} a & b & c \\ \vdots & & \\ d & e & f \end{bmatrix} * \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} a*x+b*y+c*z \\ \vdots \\ d*x+e*y+f*z \end{bmatrix}$$

as long as the number of columns matches the number of elements in the vector we can apply this multiplication to arbitrary factors. Vector Matrix multiplication is just a special case of Matrix Matrix multiplication, but we do not need to cover this now a3_m1_v1_ex1_3

So let’s implement a vector matrix multiplication using numpy. We start with the same vector a and b and turn a into a matrix by using a two dimensional numpy array. Now we can use the same overloaded dot method in order to execute the matrix vector multiplication. Note that the result is the same as two sequential vector vector dot products. But now this computation is expressed in a single method call which can easily parallelized by SIMD instructions on CPUs or by GPUs.

Deep Feedforward Networks

so this was enough math for now and should enough to understand most of neural networks math, so let’s get started with our first neural network