

# Architecting and Engineering Main Memory Database Systems in Modern C



*Chapter 0  
Organization & Introduction*

Marcus Pinnecke, M.Sc.

Research associate / Working Group Database & Software Engineering  
Institute of Technical and Business Information Systems (ITI)



# Lecturers

Marcus Pinnecke, M.Sc.

Room G29 125

Website: <http://www.dbse.ovgu.de/Mitarbeiter/Marcus+Pinnecke.html>

E-Mail: [pinnecke{at}ovgu.de](mailto:pinnecke{at}ovgu.de)

The course is  
held in english!

# Tutors

Gabriel Campero Durand, M.Sc.

Room G29 125

Website: <http://www.dbse.ovgu.de/Mitarbeiter/Gabriel+Campero+Durand.html>

E-Mail: [campero{at}ovgu.de](mailto:campero{at}ovgu.de)

Mahmoud Mohsen, B.Sc.

E-Mail: [mahmoud.mohsen{at}st.ovgu.de](mailto:mahmoud.mohsen{at}st.ovgu.de)

# PREFACE

---

Face it: you learn  
basics in your  
studies!

basics  
stuff

you are here

most industry is here

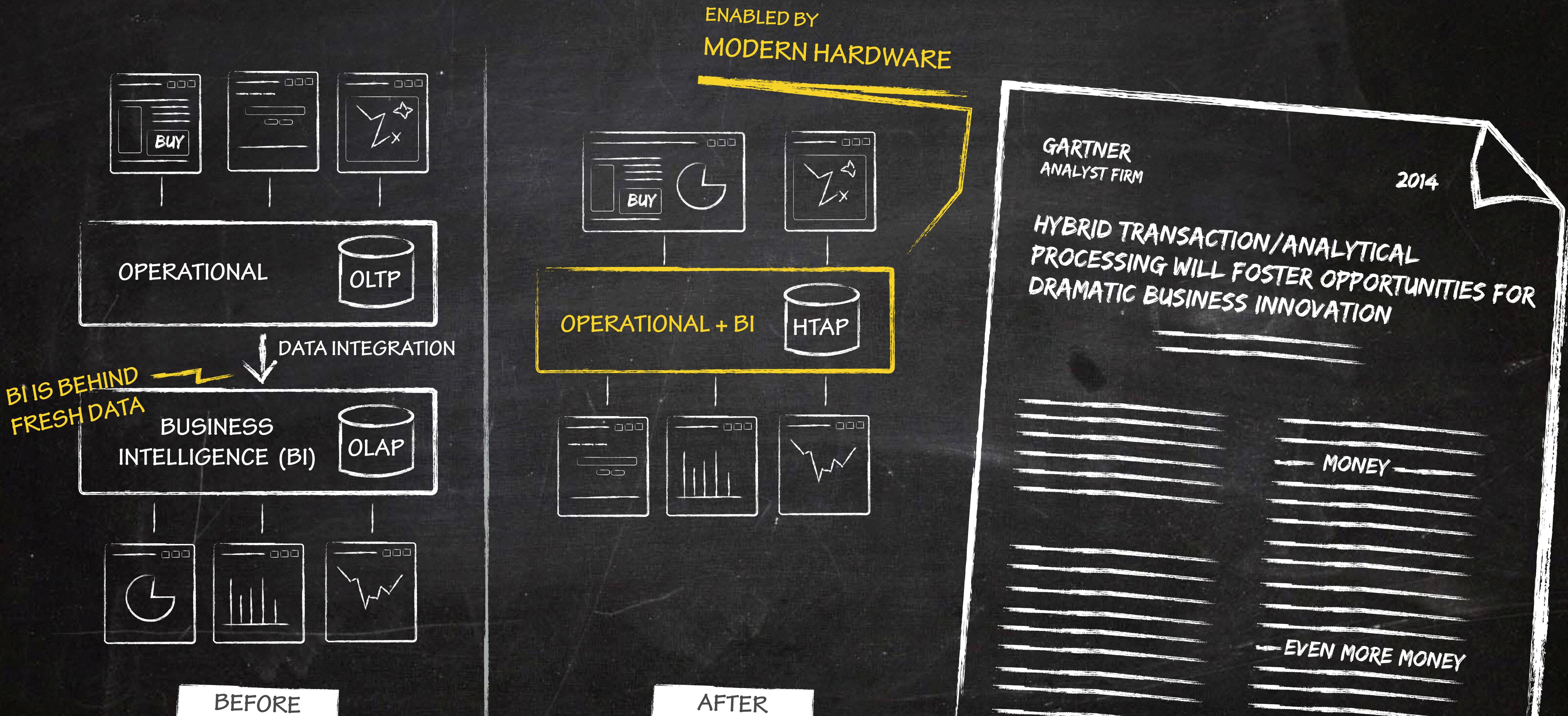
cutting-edge research  
happens here

ridculously  
advanced  
stuff that almost  
no one understands

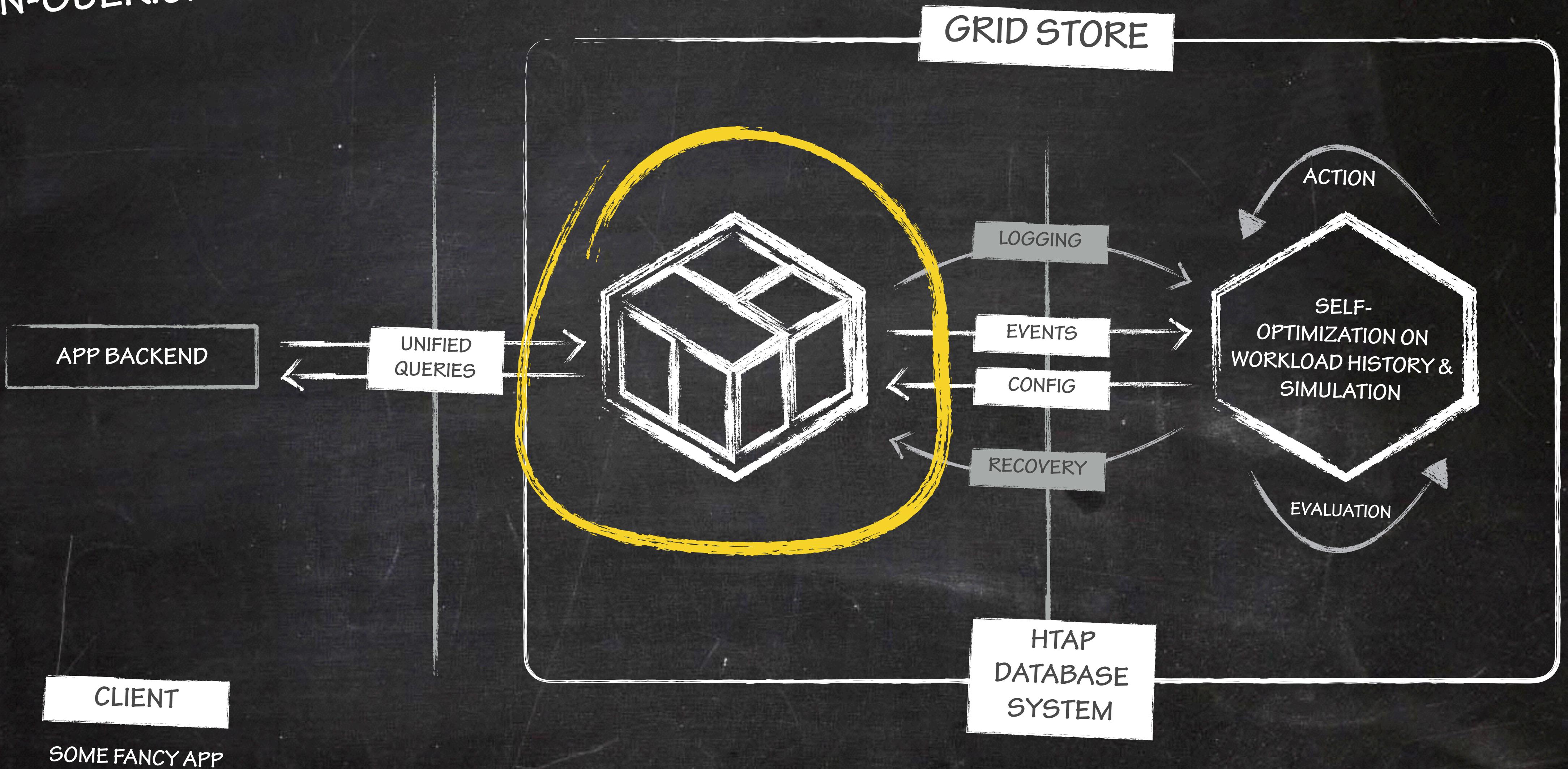
you want to be here

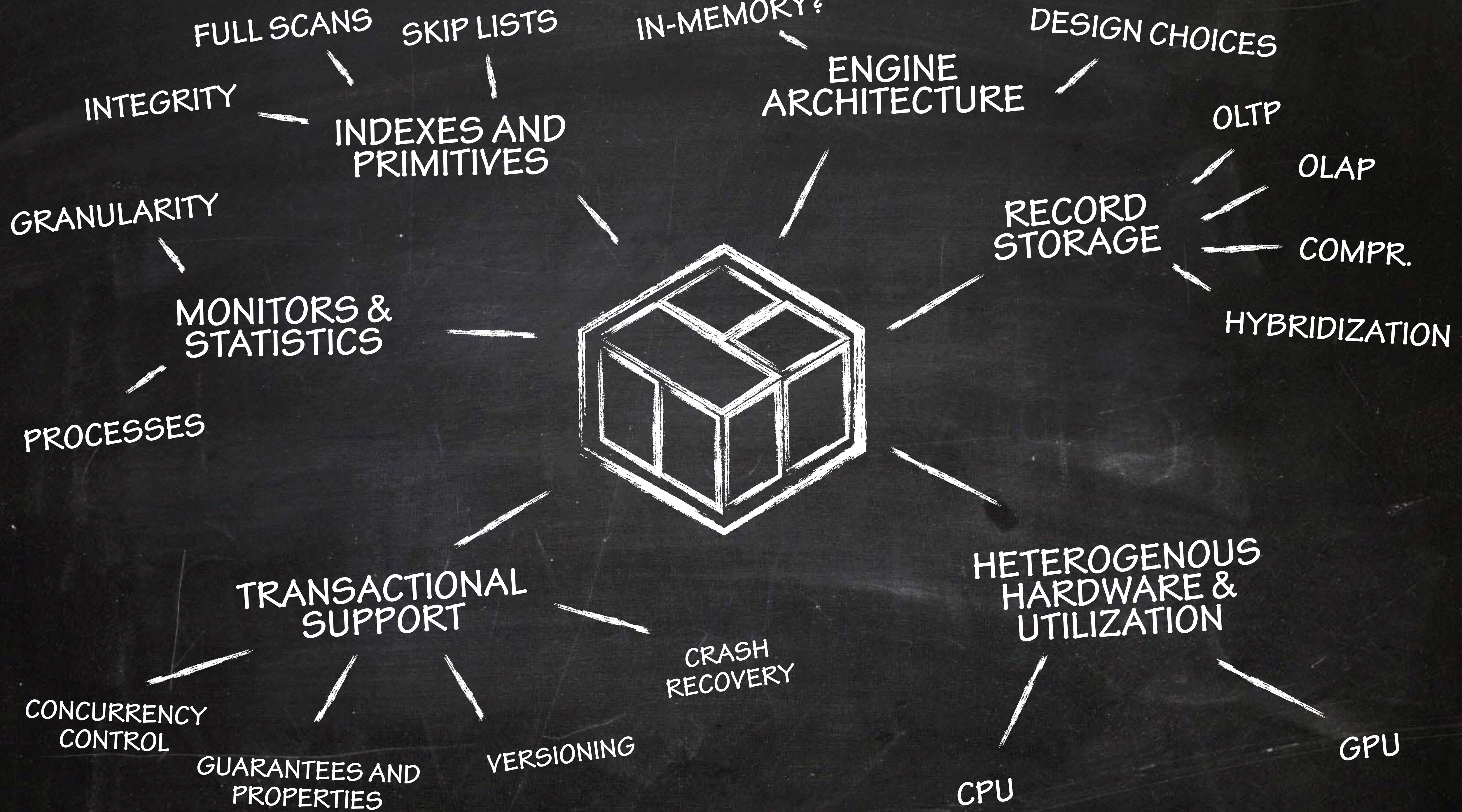
NOVEL KIND OF IN-MEMORY PLATFORM OUTPERFORMING

- ONLINE TRANSACTION PROCESSING (OLTP), AND
- ONLINE ANALYTICAL PROCESSING (OLAP)

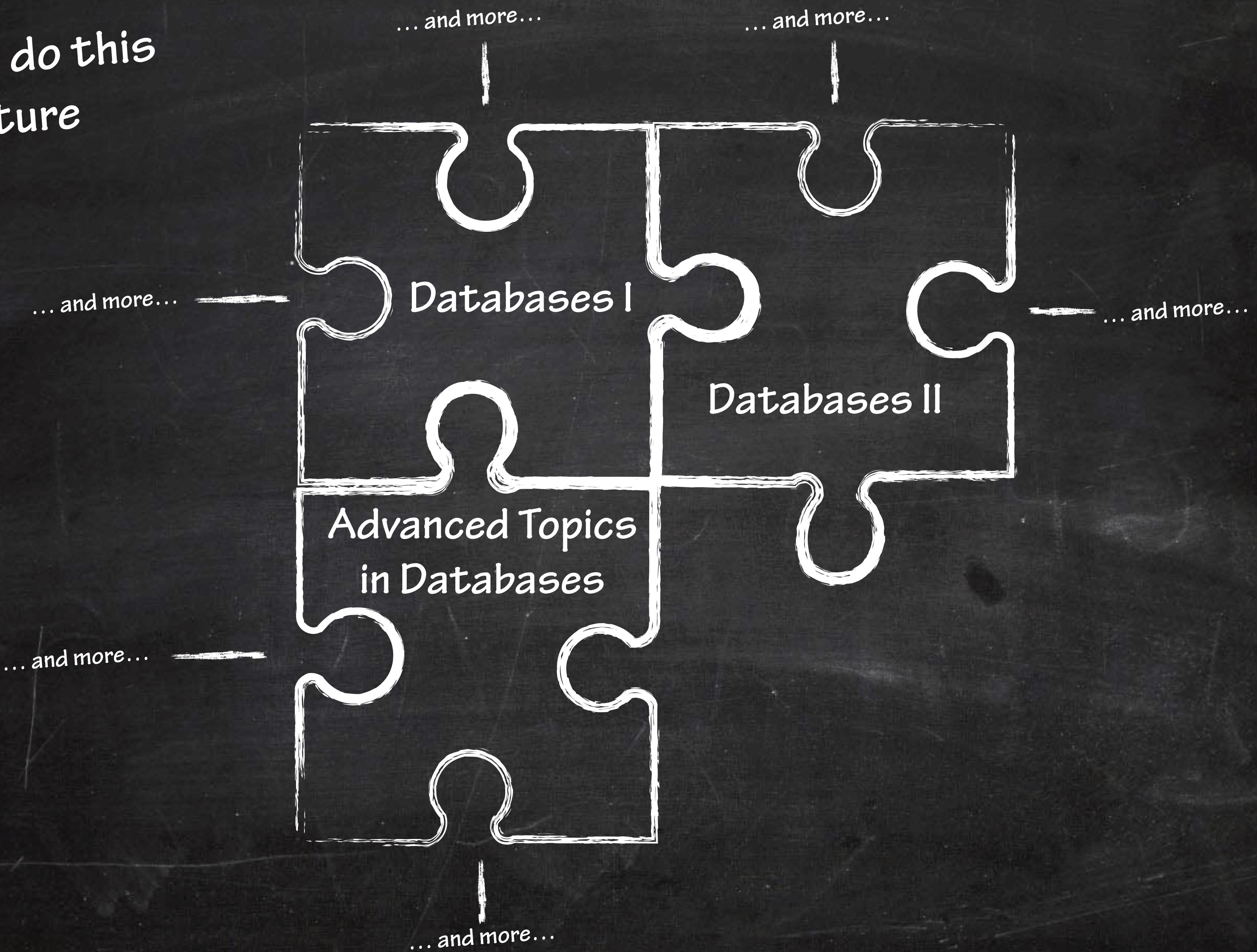


# OUR RESEARCH AT OTTO-VON-GUERICKE UNIVERSITY

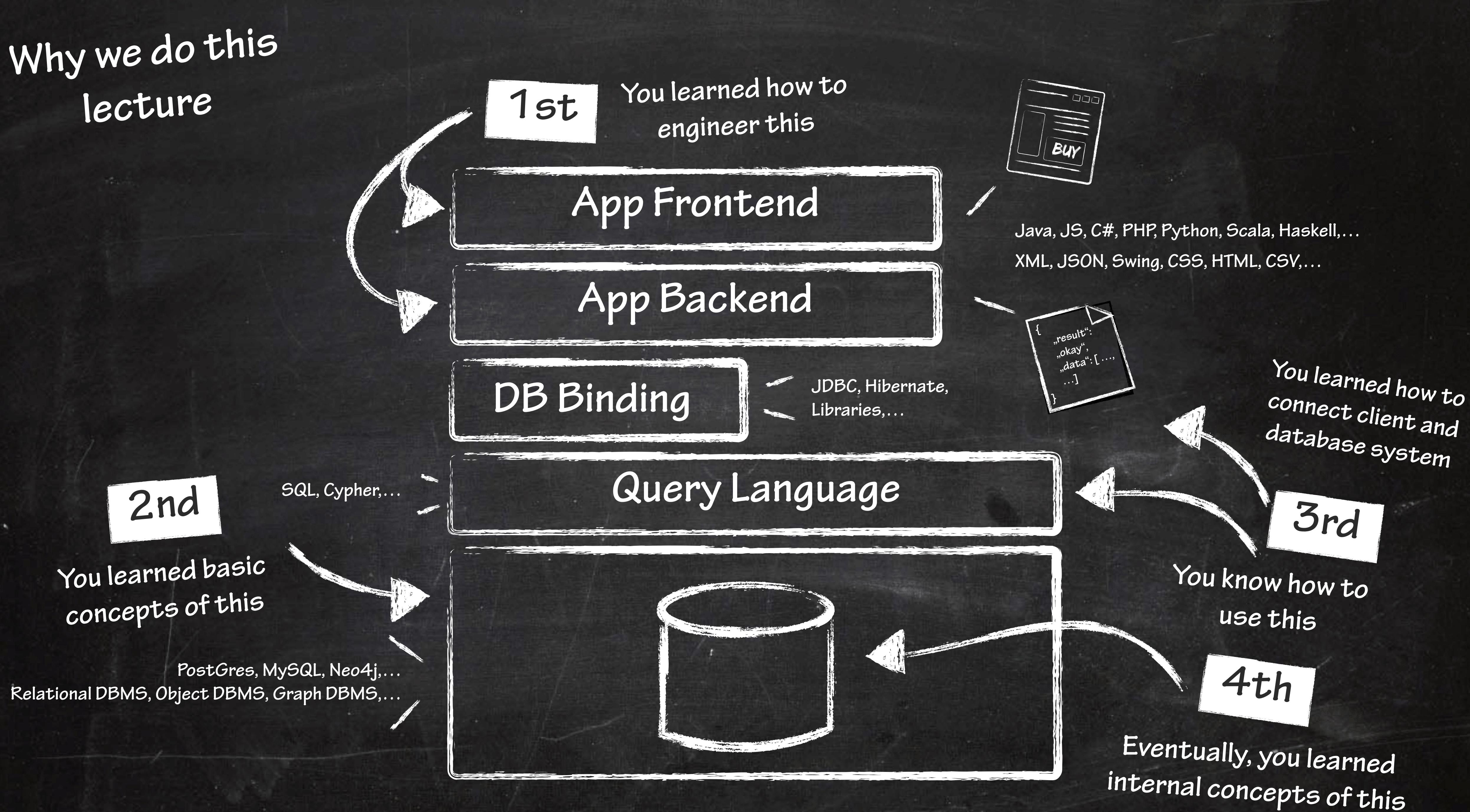




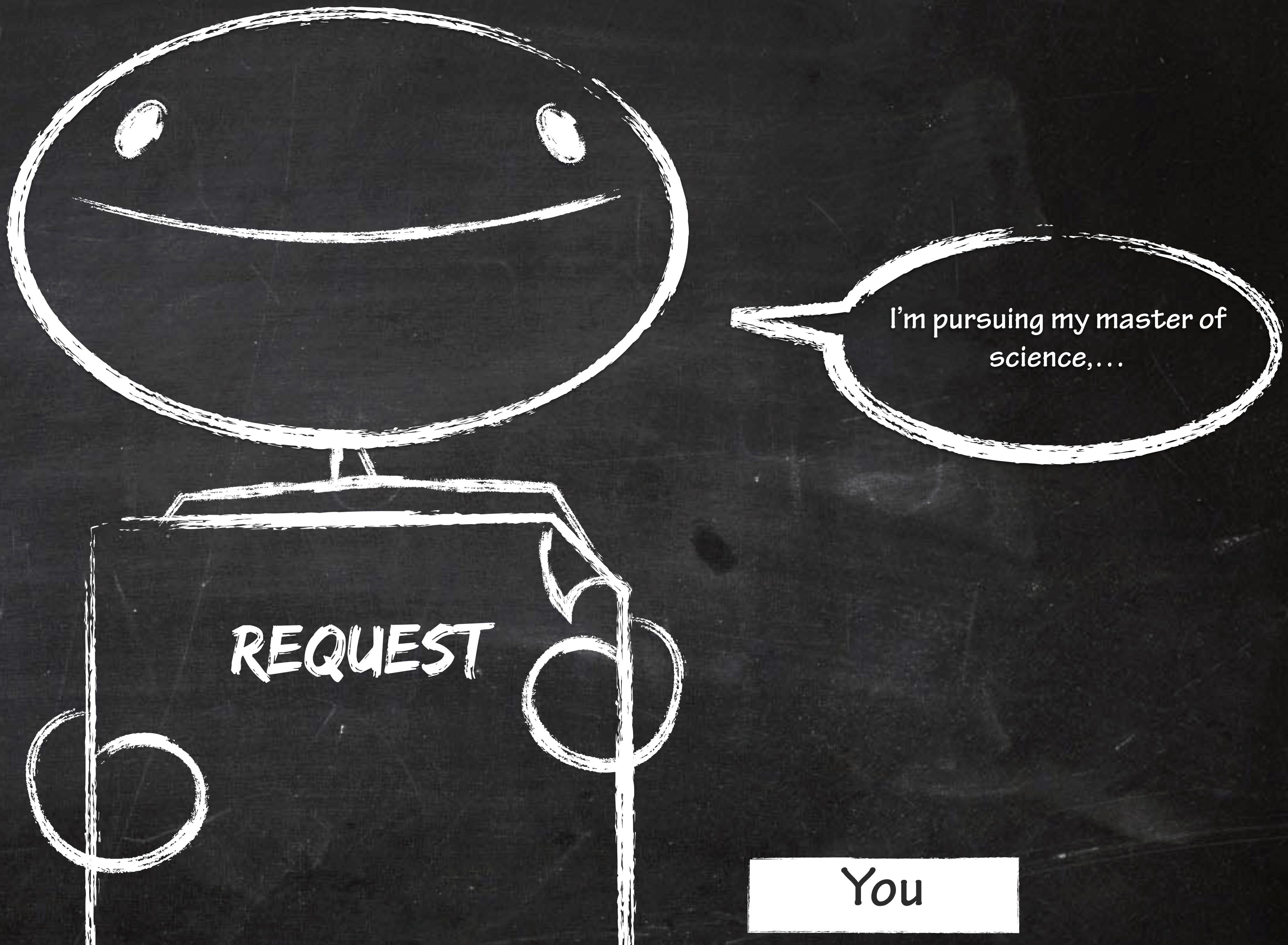
Why we do this  
lecture



# Why we do this lecture



Why we do this  
lecture

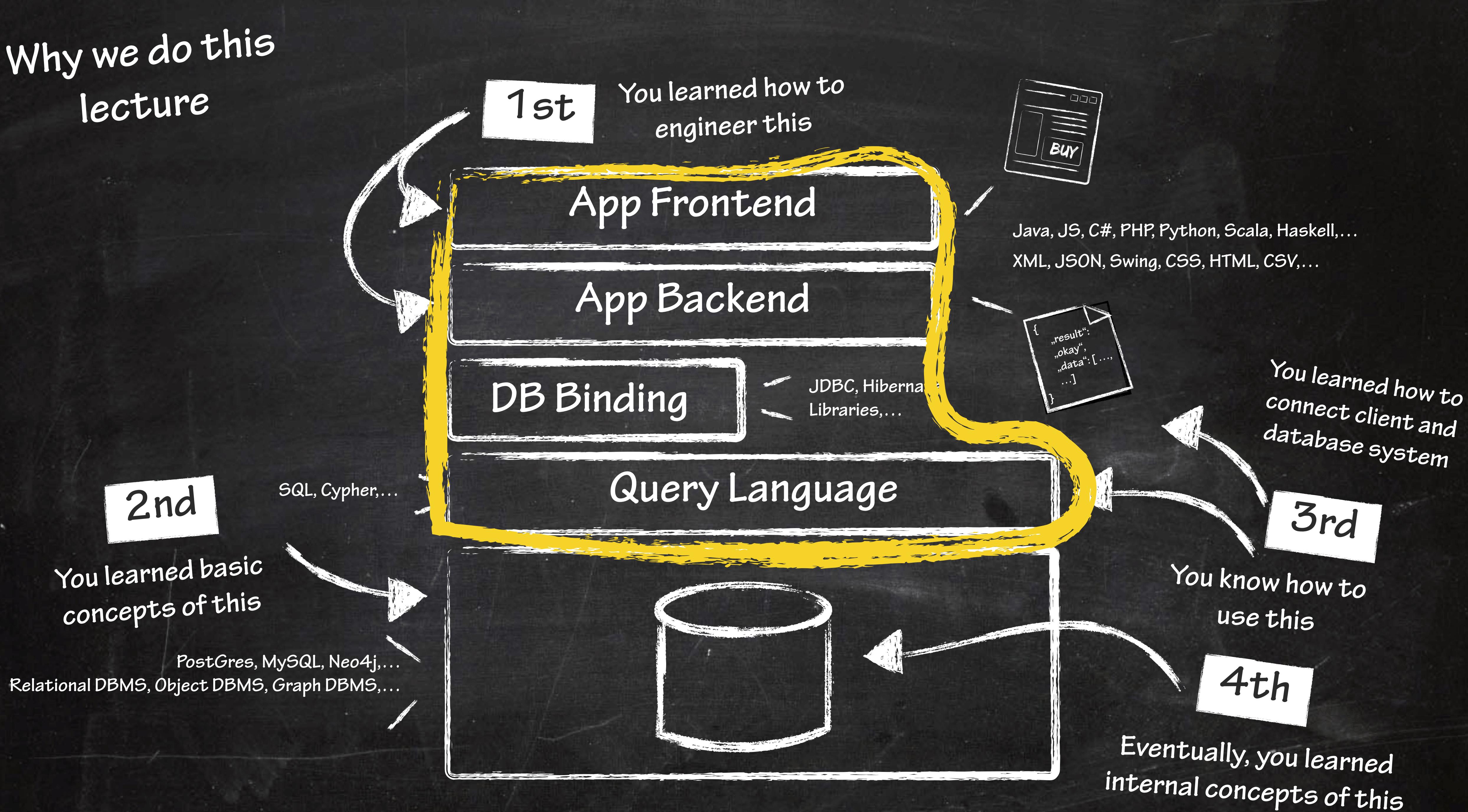


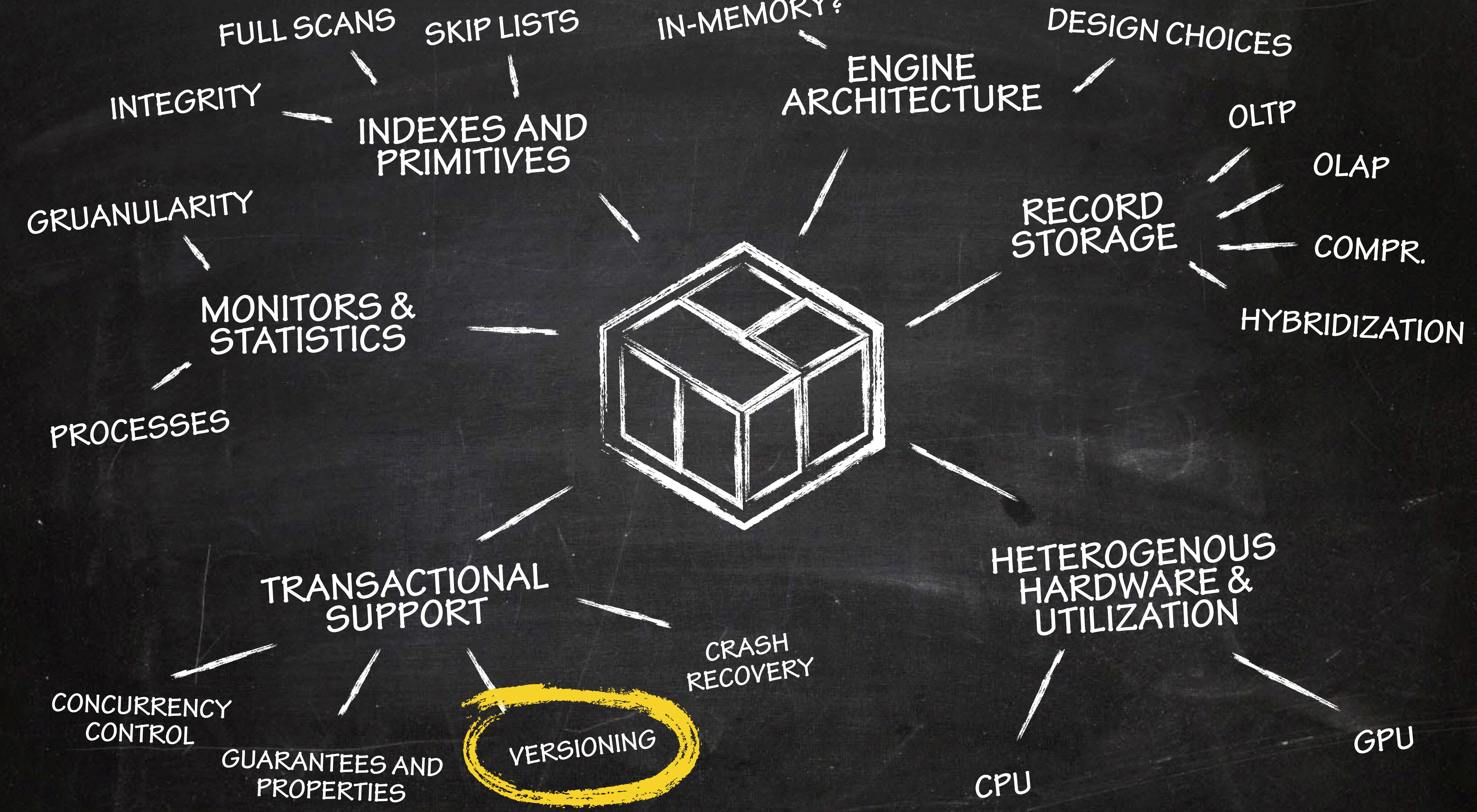
Why we do this  
lecture



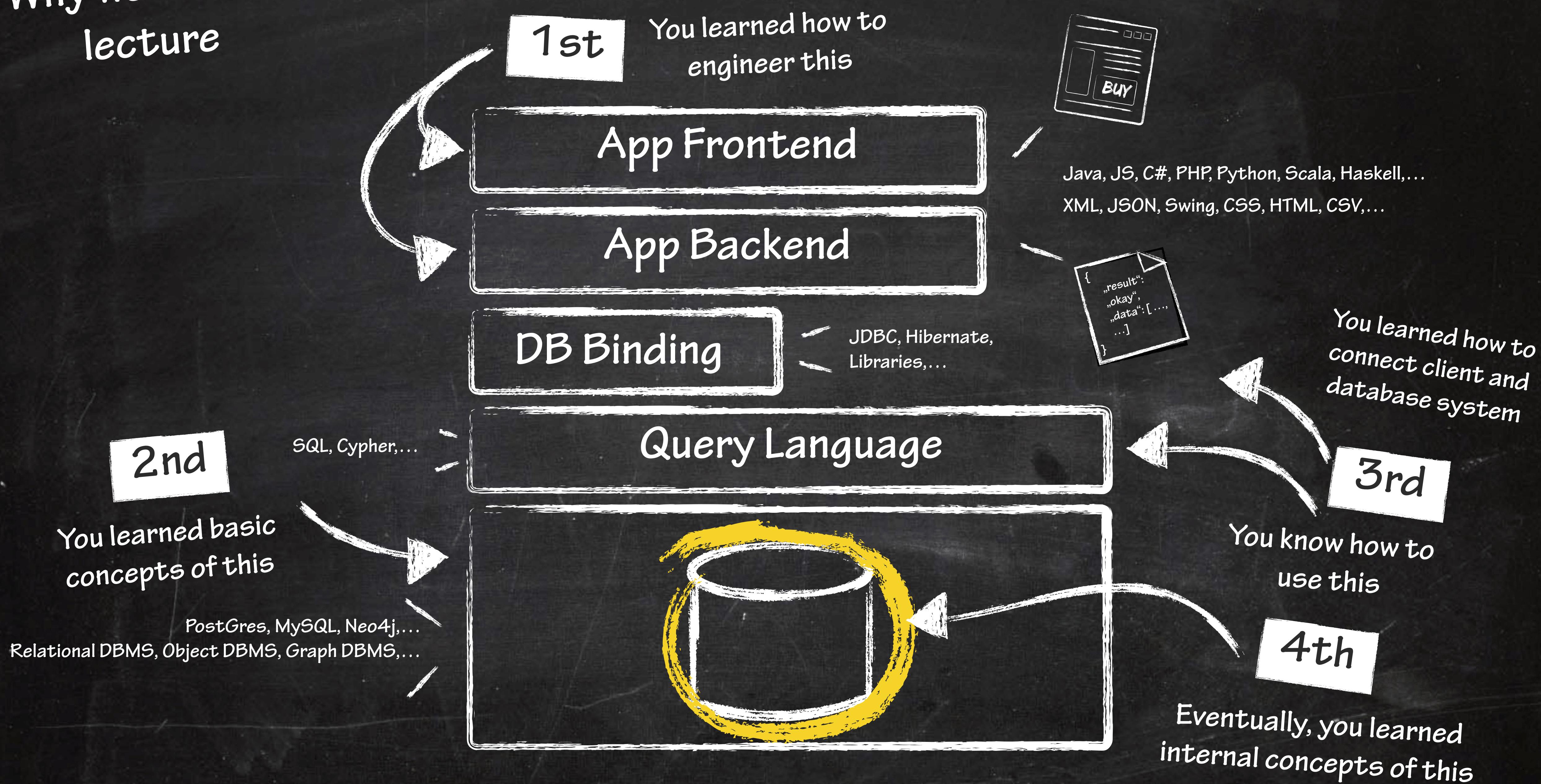
Supervisor

# Why we do this lecture





# Why we do this lecture



# Why we do this lecture

you have not much time to do the job and to understand the state of the art

you don't know the language

you don't know the standard functions

you know „only“ the basics of the domain

you rarely know how to avoid pitfalls

```
#define FLAG_NULLABLE 1 << 3

#define DELEGATE_CALL(instance, fun)
({
    REQUIRE_NONNULL(instance);
    REQUIRE_IMPL(instance->fun);
    instance->fun(instance);
})

static inline void field_update(tuplet_field_t *field, const void *data) {
    assert(field && data);
    const attr_t *attr = schema_attr_by_id(field->tuplet->fragment->schema, field->att
    if(attr_isstring(attr)) {
        const char *str = *(const char **) data;
        strcpy(field->attr_value_ptr, str);
    } else memcpy(field->attr_value_ptr, data, tuplet_field_size(field));
}
```

you don't know the toolchain

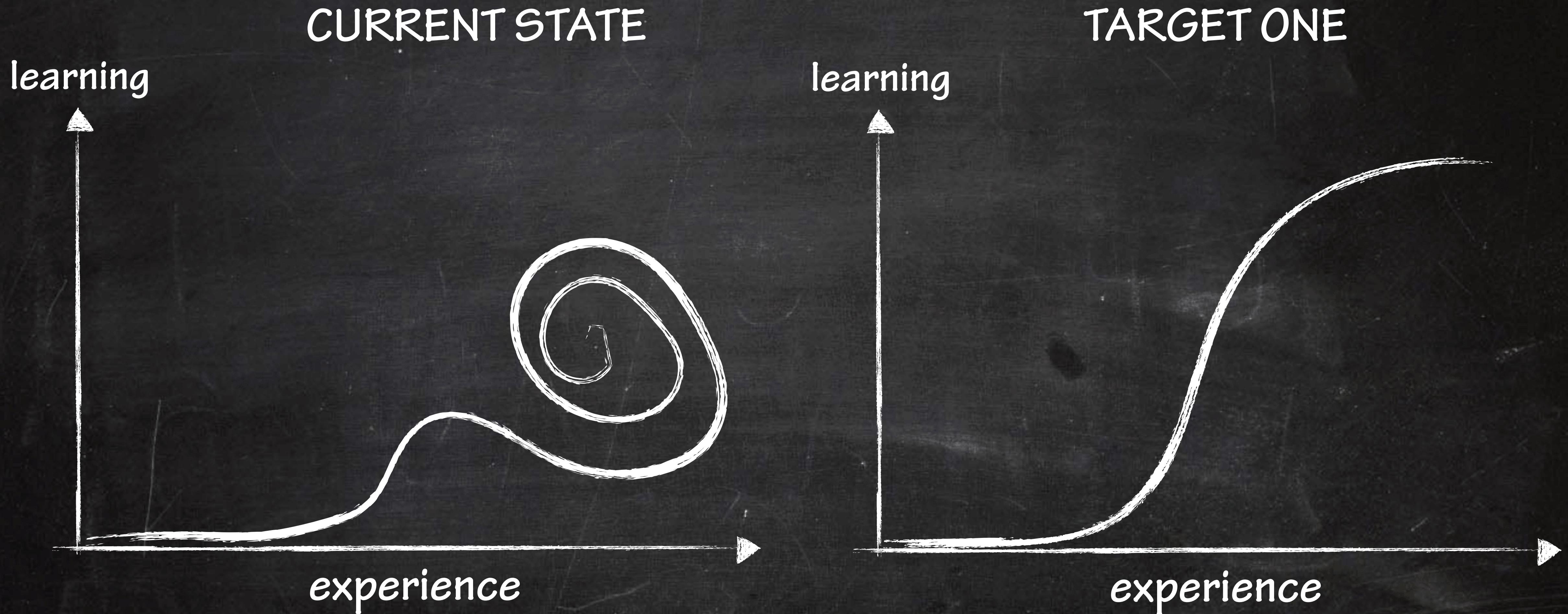
you might never get in touch w/ such a large code base

you don't know best practise

you don't know performance knobs

Why we do this  
lecture

## LEARNING CURVE



# Why we do this lecture

App Frontend

App Backend

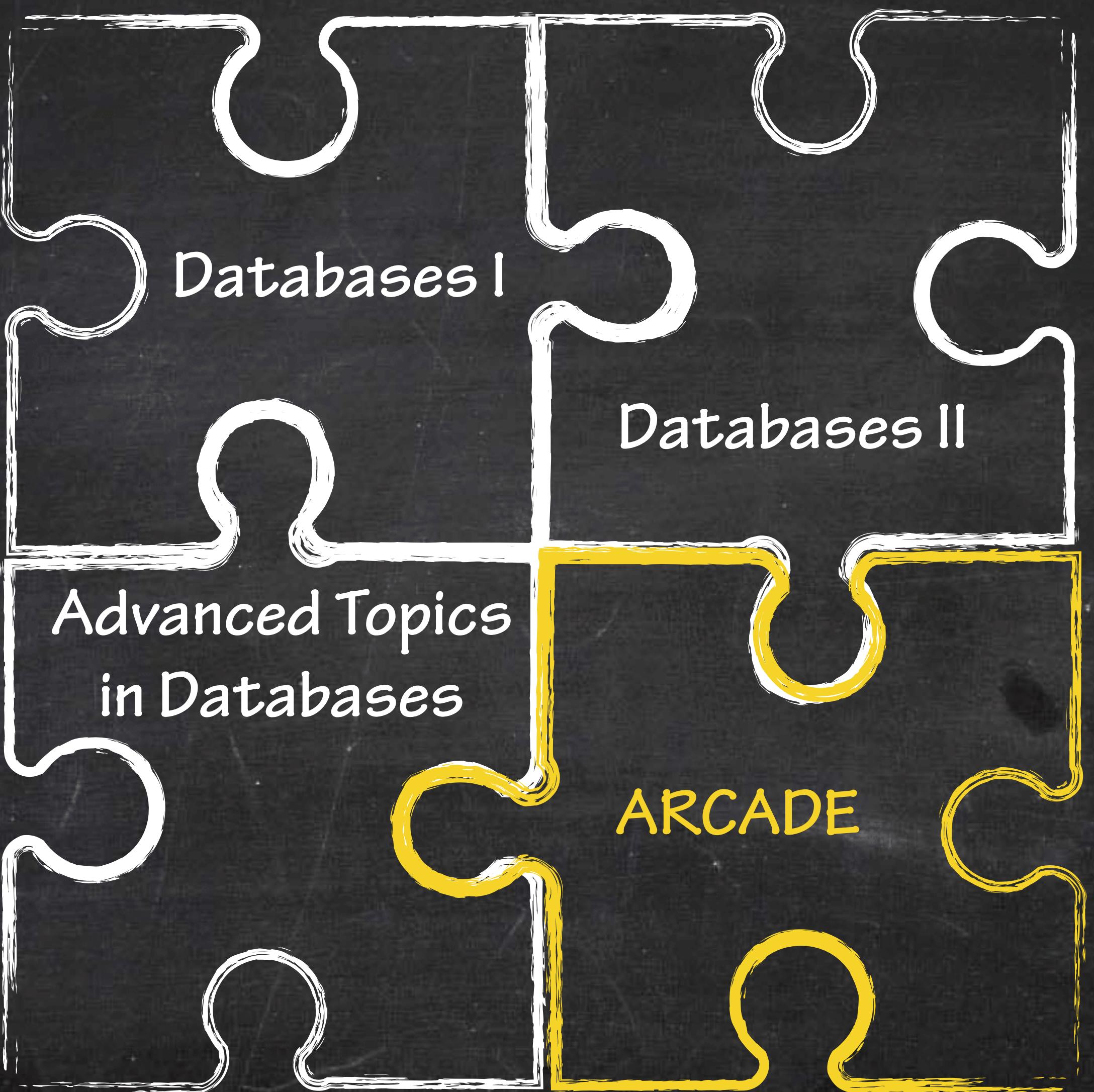
DB Binding

Query Language

In this lecture you  
will learn the black  
magic of architecting  
and engineering one  
of these



Why we do this  
lecture



# ACKNOWLEDGEMENTS

# Thanks to

Prof. Dr. Bernhard Seeger, full professor at Philipps University Marburg & chair of AG Datenbanksysteme  
... for his slides to „Programming in C++“ and also for the mentorship through the years

Dr. Sebastian Breß, researcher at German Research Center for Artificial Intelligence (DFKI)  
... for all his guidance on database systems internals (go CoGaDB go!), and his focus on co-processors

Dr. Thomas Neumann, full professor at Technical University Munich & chair of Datenbanksysteme  
... for fruitful discussion and sharing best practices on teaching system-level programming to students

Dr. Gunter Saake, full professor at Otto-von-Guericke University Magdeburg and chair of AG DBSE  
... for his support to bring novel ideas into existence and his trust into his employees

Gabriel Campero Durand, M.Sc., research associate at our working group Database & Software Engineering  
... for „being the most collaborative person (one) will ever meet“, his support for this lecture, and drive

David Broneske, M.Sc. and Andreas Meister, M.Sc. for fruitful discussion & help with administrative tasks.

Last but not least, thanks to my wife Pinni for being so pinnifull.

ORGANIZATION

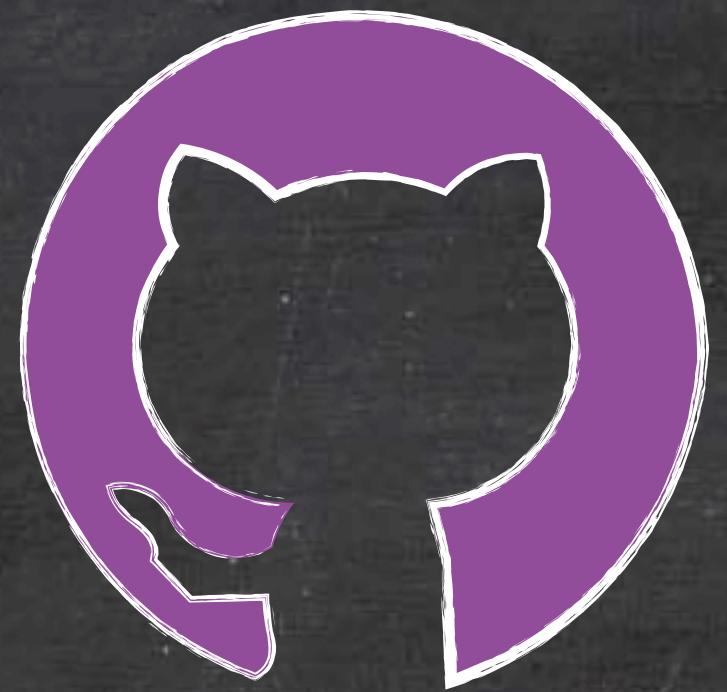
# Course Materials and Sources



facebook

NEWS,  
DISCUSSION,...

[https://www.facebook.com/  
Arcade-479499219098113/](https://www.facebook.com/Arcade-479499219098113/)



GitHub

SLIDES AND  
MATERIALS

<https://github.com/Arcade-Lecture>



GitHub

YOUR EXERCISE  
SUBMISSIONS

Up to you

# About the Course

Schedule Time

Lecture Part



11:00am to 1:00pm



Schedule Time

Exercise Part („Tutorium“)



11:00am to 1:00pm



Location  
G05-312

Material  
Slides-Only

Make notes!

Location  
G22A-113

Start  
23rd Oktober

The course is held in english!

# About the Course

Schedule Time

Lecture Part

11:00am to 1:00pm



Schedule Time

Exercise Part („Tutorium“)

11:00am to 1:00pm



Location  
G22A-113

Start  
23rd Oktober

The course is  
held in english!

Location  
G05-312

Material  
Slides-Only

Make notes!

# Make Notes!

WHAT YOU ARE  
ALLOWED TO DO...

*Everything else that is not not  
allowed.*

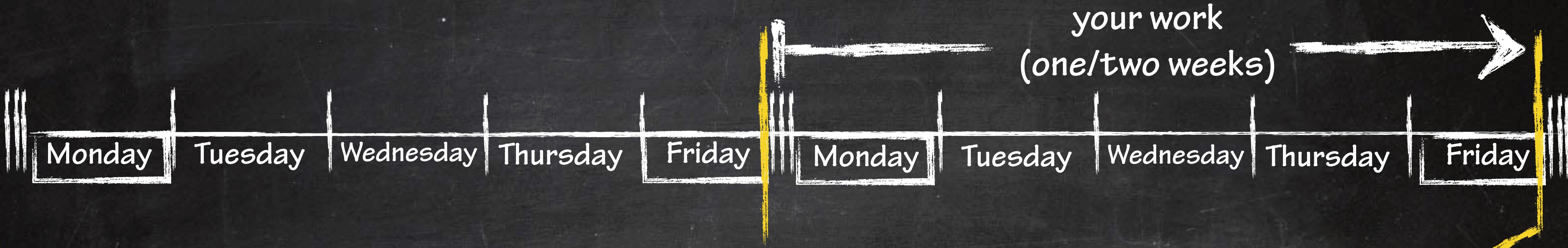
WHAT YOU ARE **NOT**  
ALLOWED TO DO...



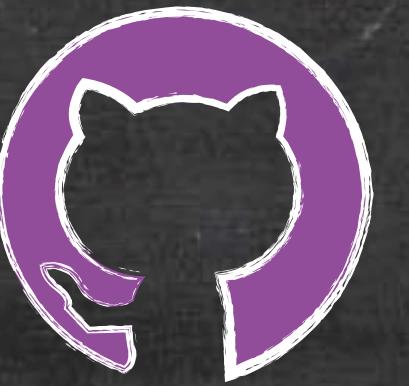
Recording

# Exercise Proceeding

## Issuing & Submission



The first sheet  
will be published  
today



Issuing

Each Friday during the day the  
exercise sheet will be published

<https://github.com/Arcade-Lecture/exercises>

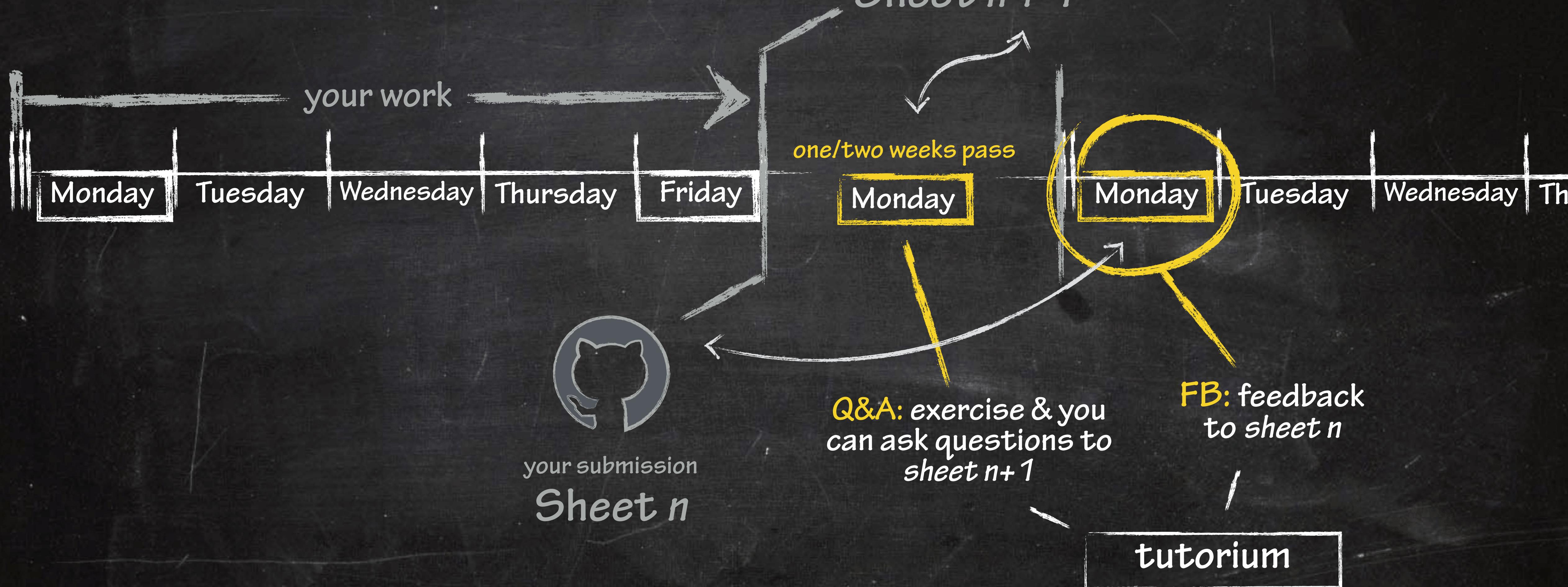


Submission

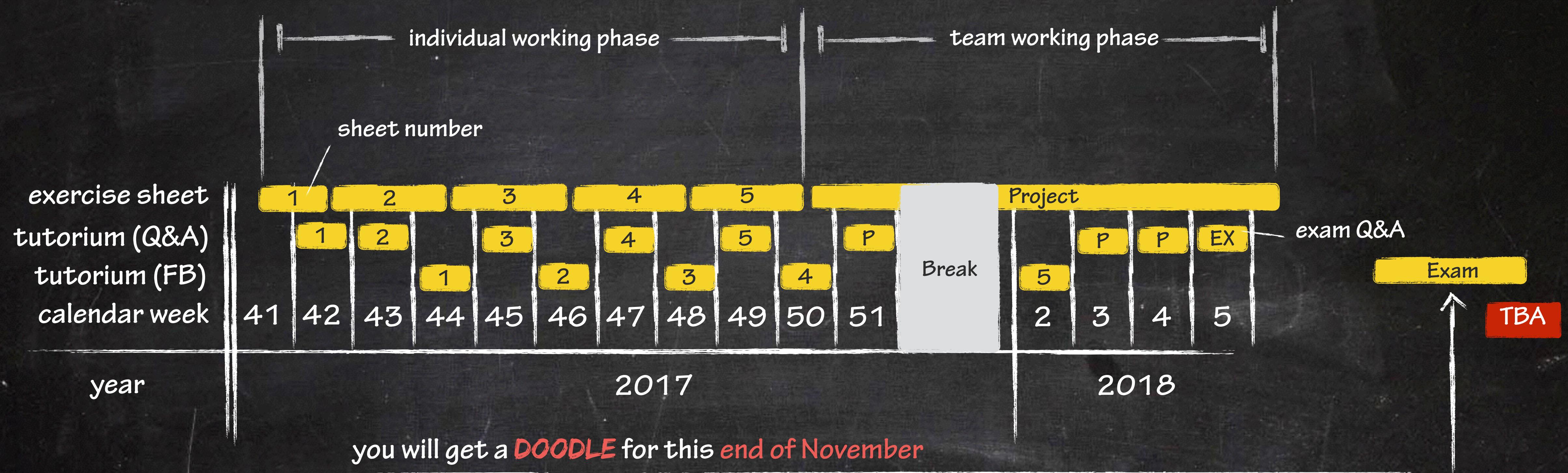
The latest change to your  
repository until 11:59pm  
of the (next) following  
Friday is your submission

# Exercise Proceeding

## Feedback & Exercise



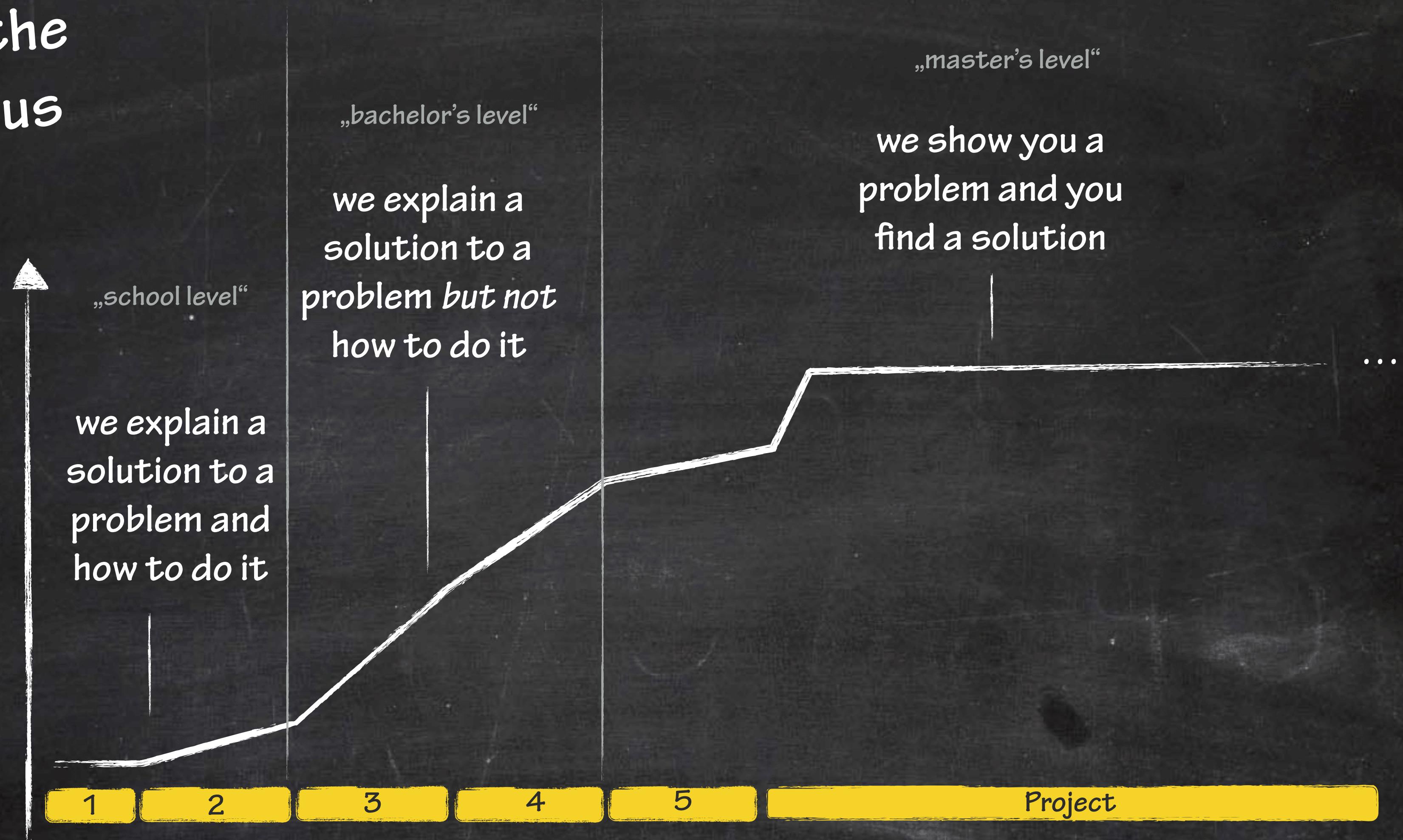
# Lecture Proceeding at a Glance



# Intention of the Exercise Focus

self-driven

guided



“ This is a picture and it should be on the wall since the wall is too naked. Take that hammer and that nail. Do the following move over and over again. ”

“ This is a picture and it should be on the wall using a hammer and some nails since the wall is too naked. ”

“ The wall is too naked. Something must be changed. ”

“ People don't like my flat. Maybe there's something wrong. ”

# Your Way to your Grade

6 CP

Note the difference  
to „volunteering“!

- manage to have **50%** of the **5** exercise sheets tasks **correct in total** (individual work).
- find **team mates** to form a group of at least two and at most  $k^*$  students. Do the **project sheet** together.
- do an **oral exam** that takes **30min**:  
**individual check of your knowledge (15min)**  
**team presentation & defense of your project (15min)**

\* the number  $k$  depends on the actual number of students participating the course.

Feel free to visit or  
not to visit the  
lecture or tutorium

# What's Next after your Grade?

you are trained in a more complex engineering process rather than „just exercising”

it's easier to find a topic that you like

- you are a better engineer than now\*
- individual and team projects
- software projects
- bachelor's/master's thesis
- ... maybe even jobs ;)

you are skilled enough to write code that remains in the code base

you're already familiarized with the system

you are more productive in a shorter time; you can do high-quality tasks

# Important Dates

1<sup>st</sup> Sheet  
Today

1<sup>st</sup> Tutorial (Q&A)  
16<sup>th</sup> Okt 17

1<sup>st</sup> Sheet Deadline  
20<sup>th</sup> Okt 17

1<sup>st</sup> Sheet Feedback  
30<sup>th</sup> Okt 17

Last Sheet Deadline / Project Start  
15<sup>th</sup> Dec 17

Exam  
TBA

*feedback is welcome!*