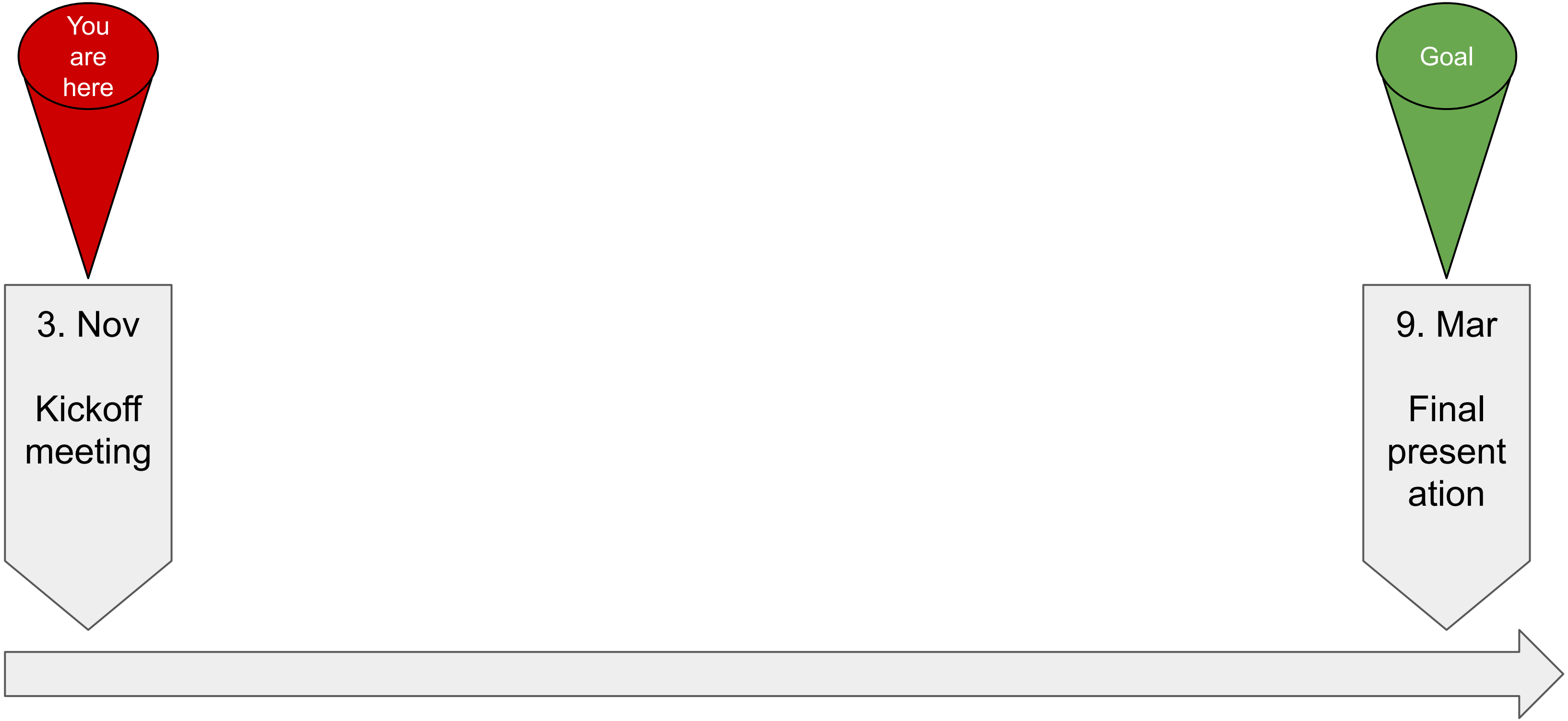
Programming Database Web Applications

Christian Winter (winterch@in.tum.de)
Alexander van Renen (renen@in.tum.de)
Prof. Alfons Kemper

Topics

1. Course Organization
2. Inspiration for Projects
3. Prospeum Pitch

The Plan



You are here

3. Nov

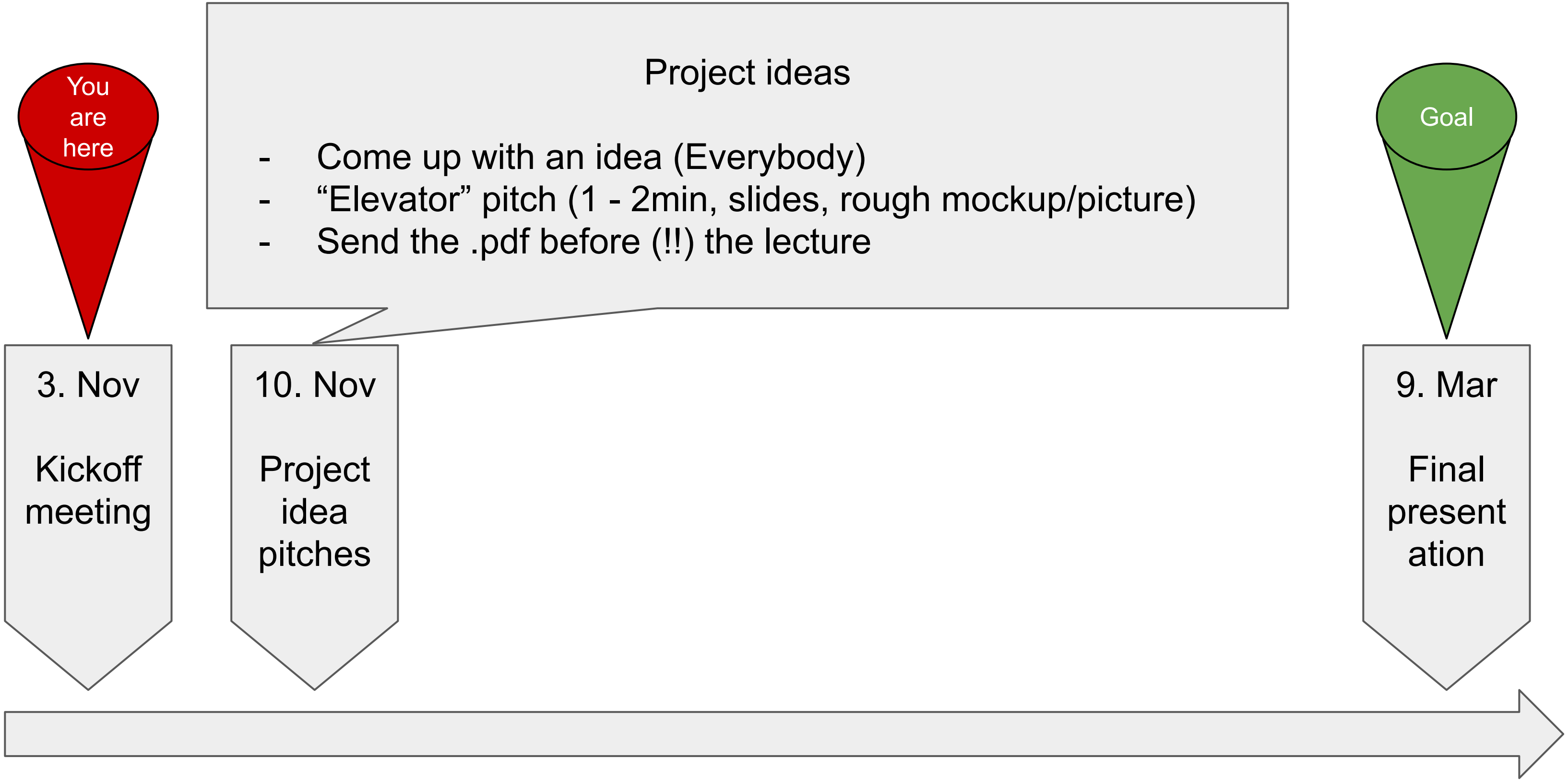
Kickoff meeting

Goal

9. Mar

Final present ation

The Plan



You are here

3. Nov

Kickoff meeting

Project ideas

- Come up with an idea (Everybody)
- “Elevator” pitch (1 - 2min, slides, rough mockup/picture)
- Send the .pdf before (!!) the lecture

10. Nov

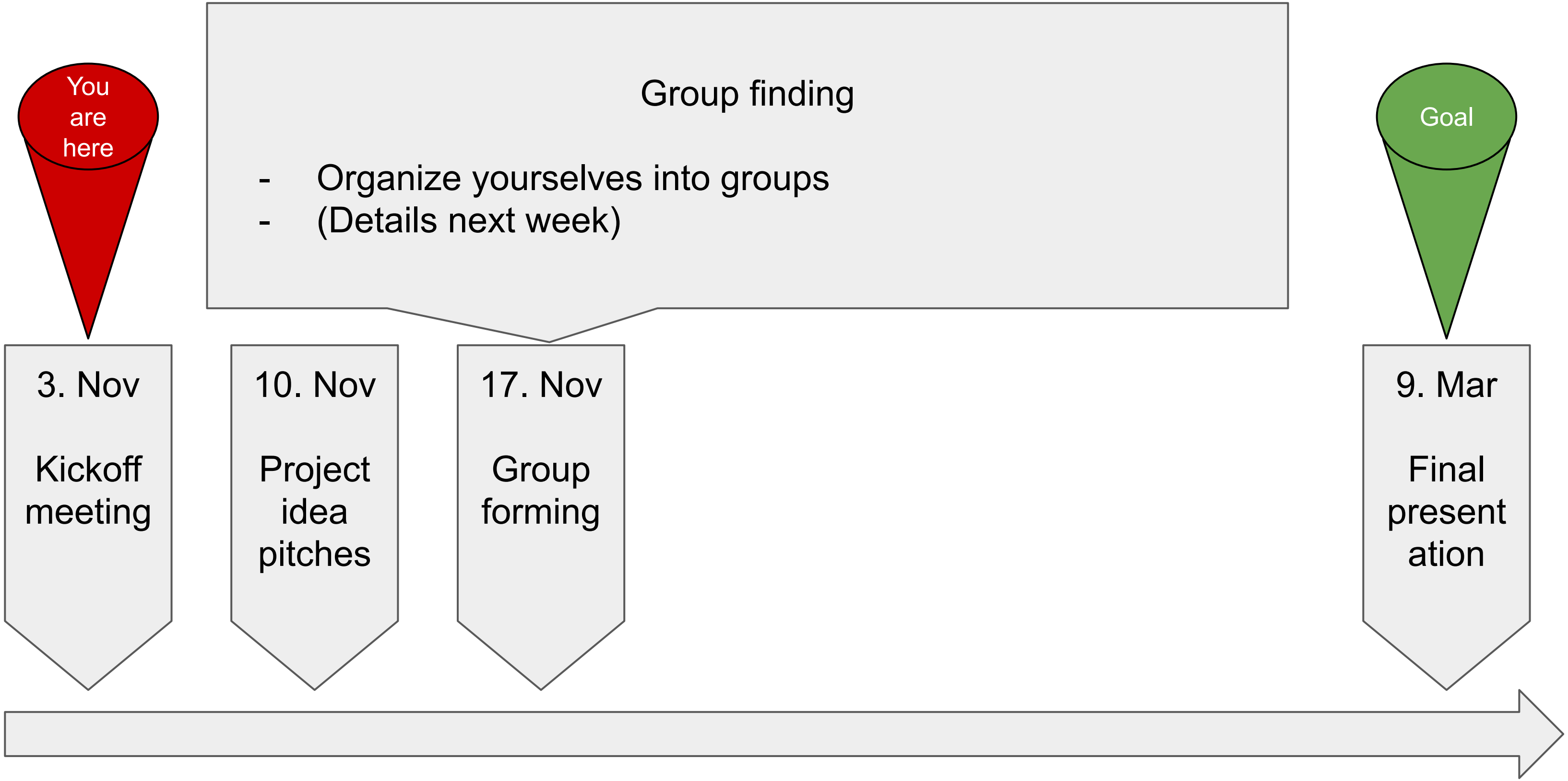
Project idea pitches

Goal

9. Mar

Final presentation

The Plan



Group finding

- Organize yourselves into groups
- (Details next week)

You are here

3. Nov

Kickoff meeting

10. Nov

Project idea pitches

17. Nov

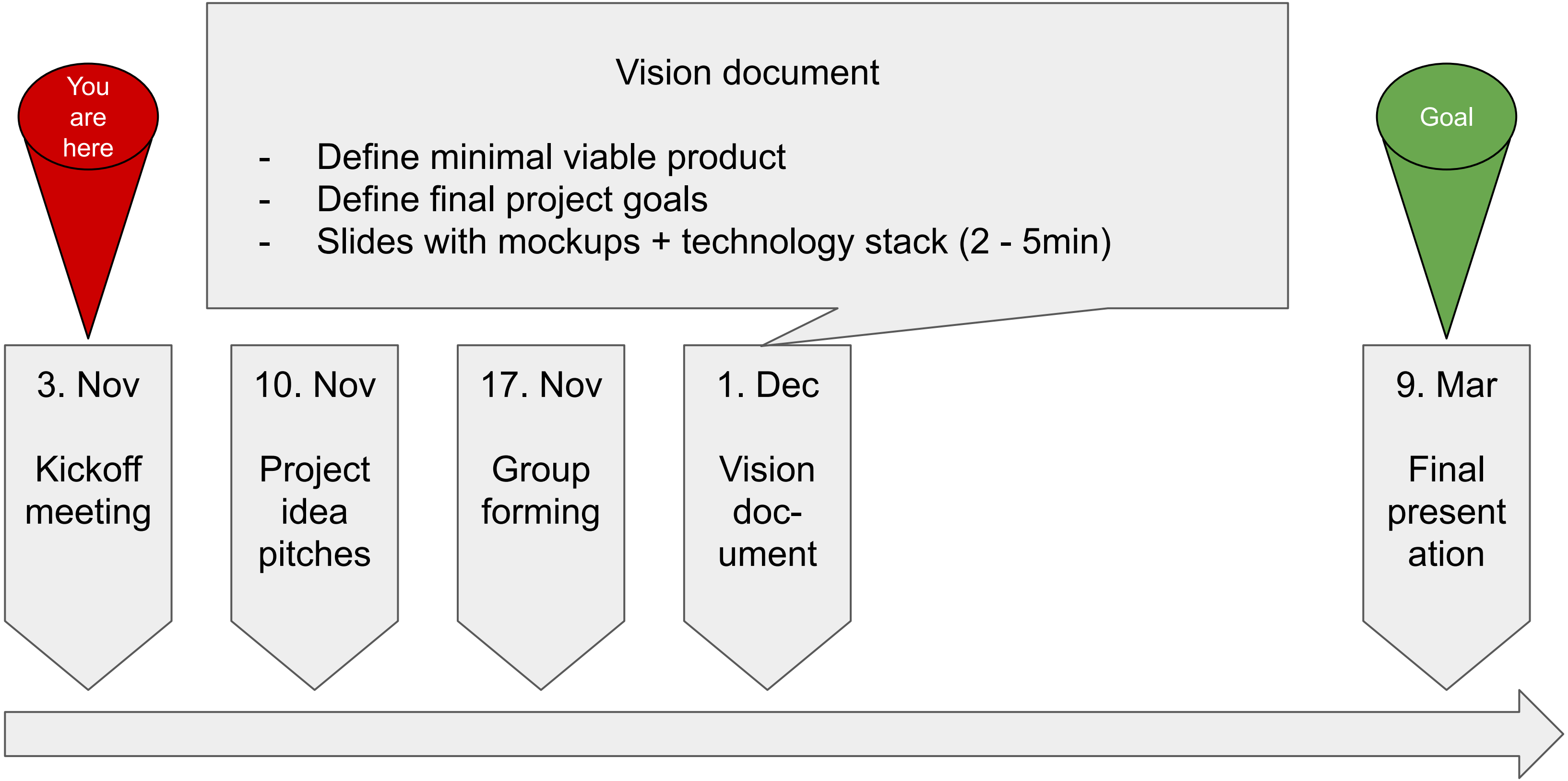
Group forming

Goal

9. Mar

Final present ation

The Plan



Vision document

- Define minimal viable product
- Define final project goals
- Slides with mockups + technology stack (2 - 5min)

You are here

Goal

3. Nov – Kickoff meeting

10. Nov – Project idea pitches

17. Nov – Group forming

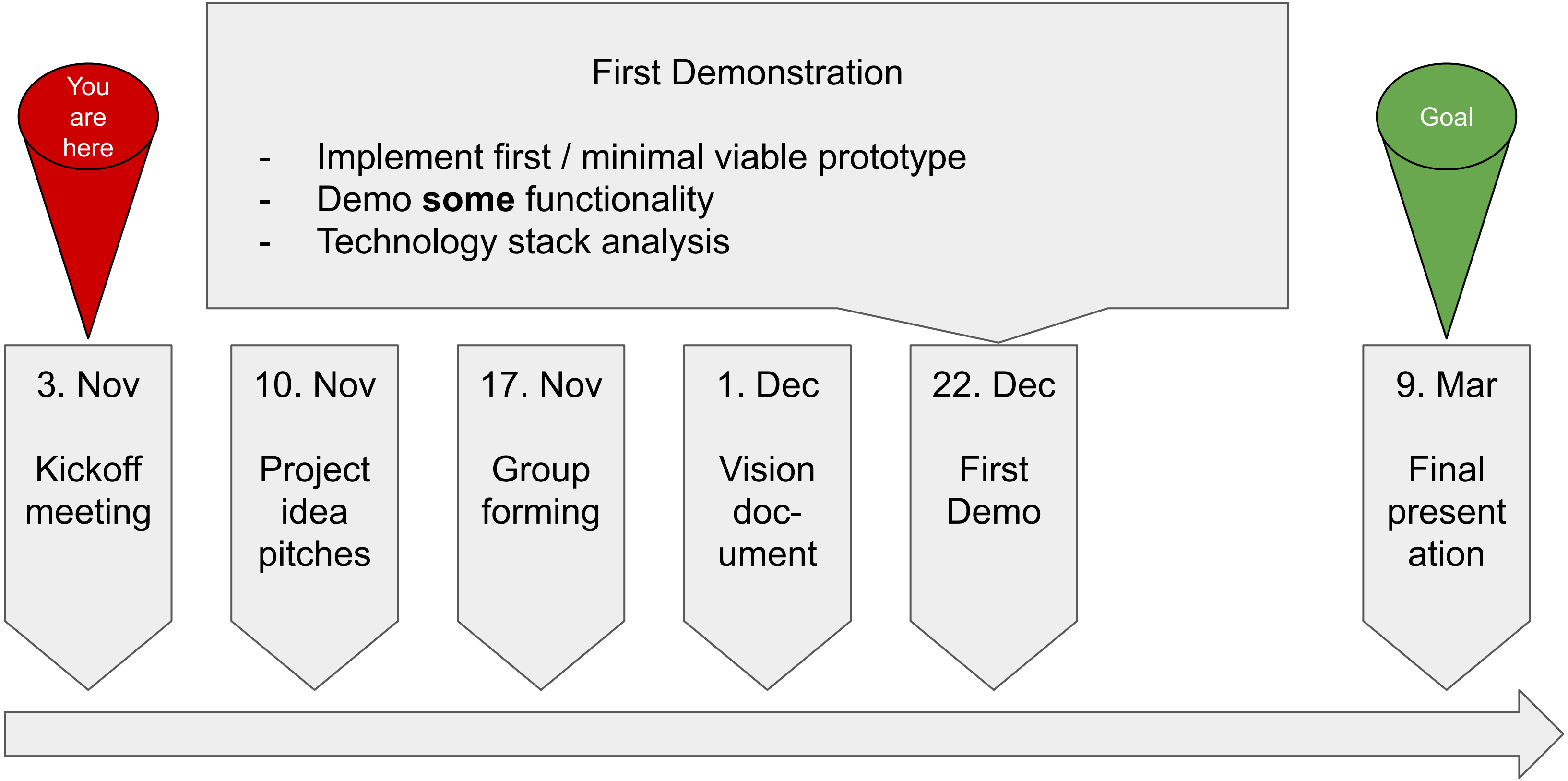
1. Dec – Vision document

9. Mar – Final presentation

Vision Document

- Similar to scope + requirements statement
- Requirements statement (“Lastenheft”):
  - Motivation + Problem description
  - Project goal (what would the final product look like)
- Scope statement (“Pflichtenheft”):
  - System Architecture
  - Technology stack
  - Project scope (what will we implement for this course)
- Roughly 2-3 pages
- Due 1. December

The Plan



First Demonstration

- Implement first / minimal viable prototype
- Demo **some** functionality
- Technology stack analysis

You are here

Goal

3. Nov

Kickoff meeting

10. Nov

Project idea pitches

17. Nov

Group forming

1. Dec

Vision doc-ument

22. Dec

First Demo

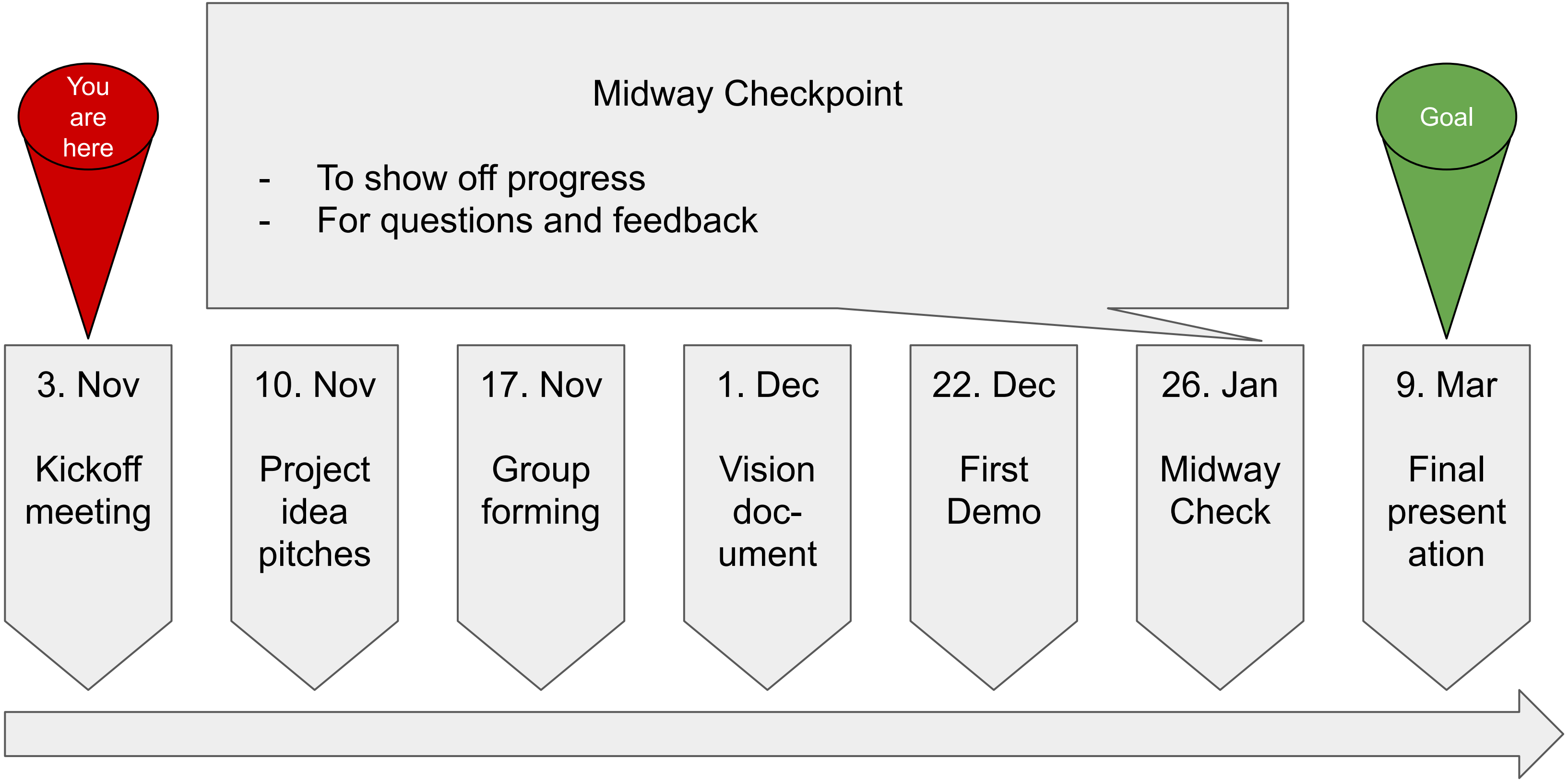
9. Mar

Final present ation

First Demonstration: MVP

- Implement the first prototype
- Demo **some** functionality
- Technology Stack Analysis:
    - Justify the choices for the stack
    - Focus on database interaction
- Lessons learnt
    - What were the problems that you faced?
    - How did you solve them?
    - How did you divide the work among you?
- 5-8 slides
- Due 22. December

The Plan



You are here

Midway Checkpoint

- To show off progress
- For questions and feedback

Goal

3. Nov

Kickoff meeting

10. Nov

Project idea pitches

17. Nov

Group forming

1. Dec

Vision doc-ument

22. Dec

First Demo

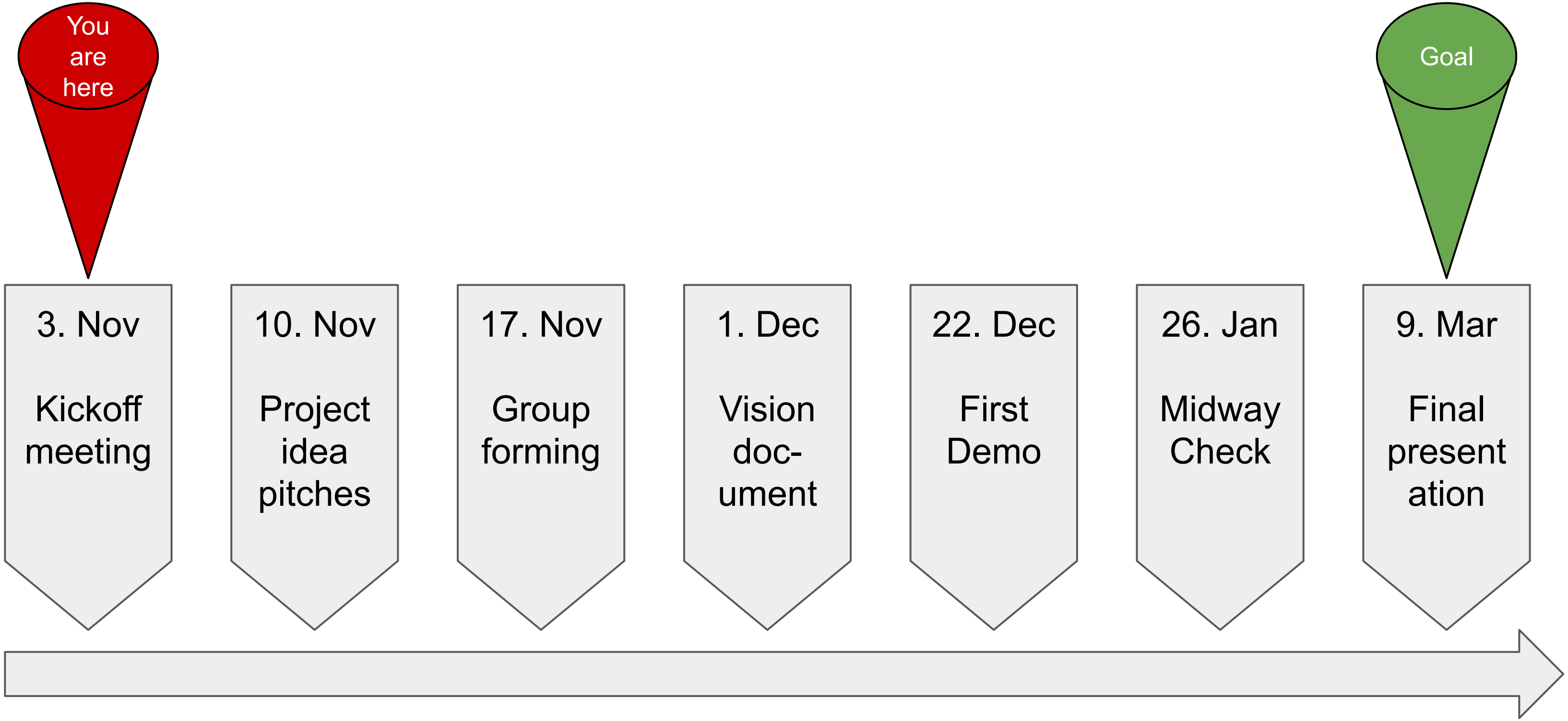
26. Jan

Midway Check

9. Mar

Final present ation

The Plan - Overview



You are here

Goal

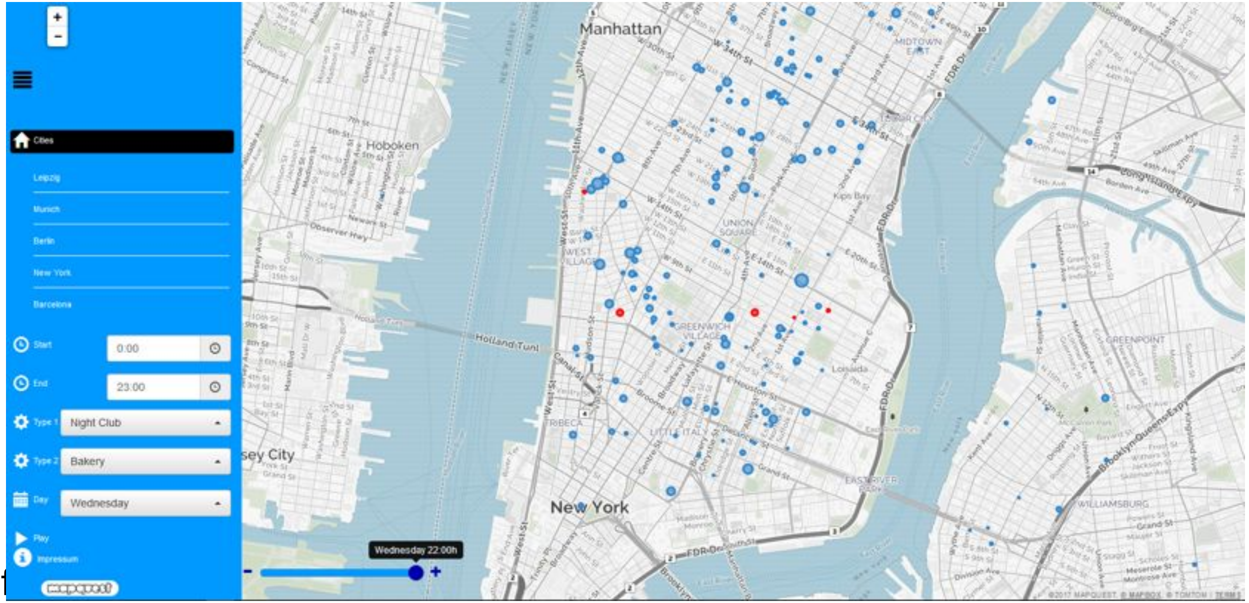
- 3. Nov: Kickoff meeting
- 10. Nov: Project idea pitches
- 17. Nov: Group forming
- 1. Dec: Vision document
- 22. Dec: First Demo
- 26. Jan: Midway Check
- 9. Mar: Final presentation

MapViz (2016)

- Popular-Times feature by Google
- Feature shows relative amount of visitors at a specific place

⇓

- Visualization of data
- Identify movement patterns of people



Start 0:00

End 23:00

Type 1 Night Club

Type 2 Bakery

Day Wednesday

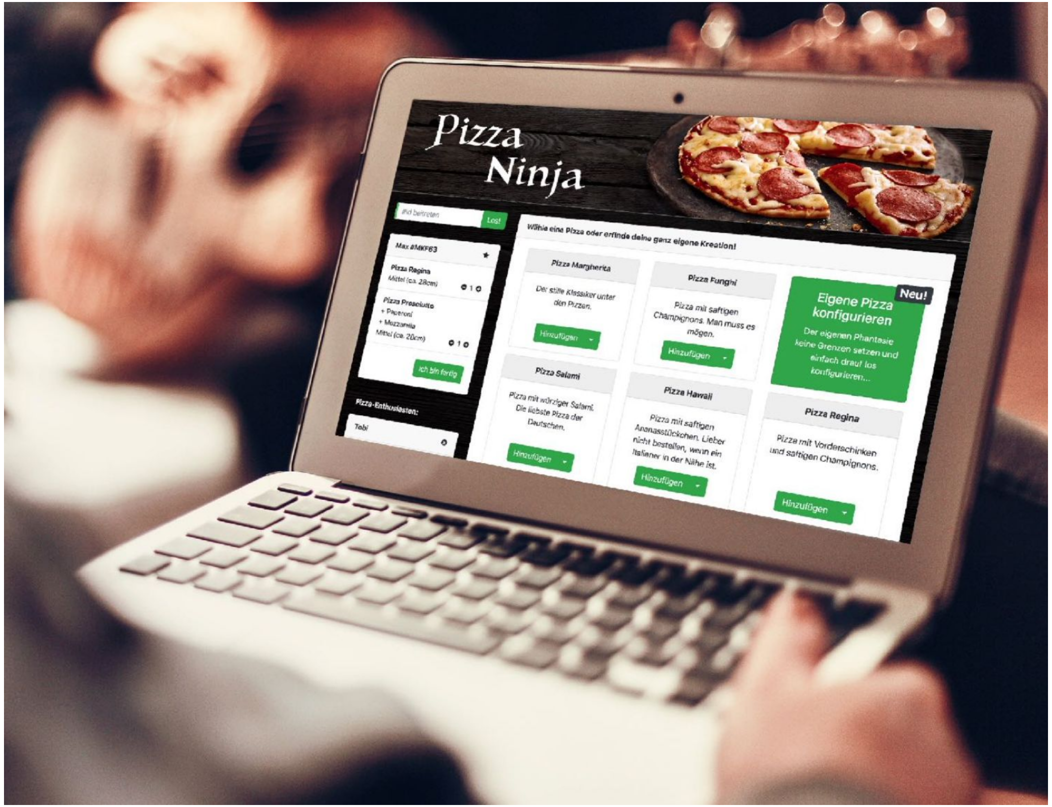
Wednesday 22:00h

Pizza Ninja (2017)

- Crawl data from pizza delivery services

⇩

- Decouple ordering from choosing a restaurant
- Order in a group



Wähle eine Pizza oder erfinde deine ganz eigene Kreation!

Der stille Klassiker unter den Pizzen.

Pizza mit saftigen Champignons. Man muss es mögen.

Der eigenen Phantasie keine Grenzen setzen und einfach drauf los konfigurieren...

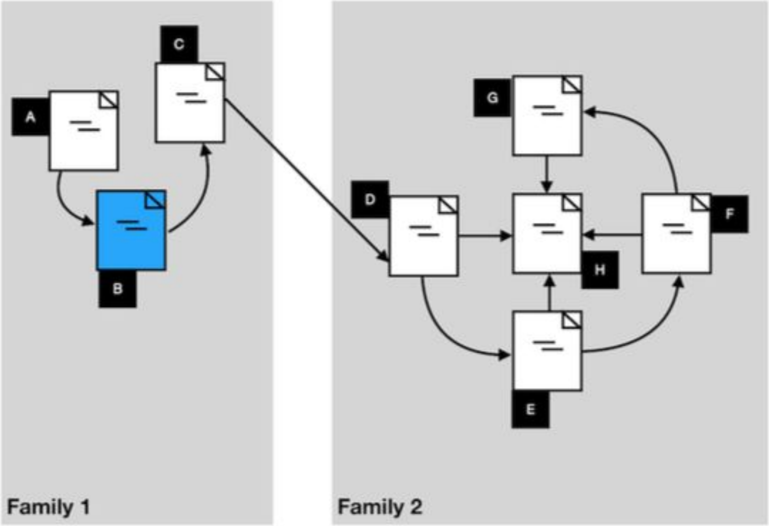
Pizza mit würziger Salami. Die liebste Pizza der Deutschen.

Pizza mit saftigen Ananasstückchen. Lieber nicht bestellen, wenn ein Italiener in der Nähe ist.

Pizza mit Vorderschinken und saftigen Champignons.

Research Graph (2019)

- Crawl online paper APIs
- Find familiarities between papers (weighted citations)
- Visualize !!!



#1 Smart Weather Forecast

Build a website to interactively explore these the differences between the forecasted weather and the actual (measured) weather.

- Allows professionals to identify weaknesses in weather models
- Detect which weather service is best for which location
- Analyse difference between forecast and measurement
- Temperature, wind, pressure ...
- Big data management: 100GB+ weather data

#2 Traffic Aware Routing

User wants to drive from A to B and wants to arrive by datetime X

When should he/she depart to minimize the travel time?

Tasks:

- Get necessary data from public data sources
- Design and create database
- Design and implement web interface
- Plot estimated travel (y axis) and departure times (x axis)

#3 phpMyAdmin for node.js

This project would allow web developers to configure their database in node js.

Tasks:

- Design and create secure management platform for a modern database.
- Could be done for postgres to help make it more popular in the node community.
- Many possibilities: Provide enhanced analytics, easy setup, management features, monitoring ...
- Contribution to open source community.

#4 Tutor Tool (!!!)

A website to help manage big lectures with multiple tutor groups.

Features:

- Existing system
- Tutored accounts, tutor groups, bonus points, export, import
- If finished (and good), it will definitely be used !

#5 Plan Ahead

US has a number of National Parks and areas which do not have cellular coverage. Also, a number national parks do not have any options for food/water at the visitor centers.

Tasks:

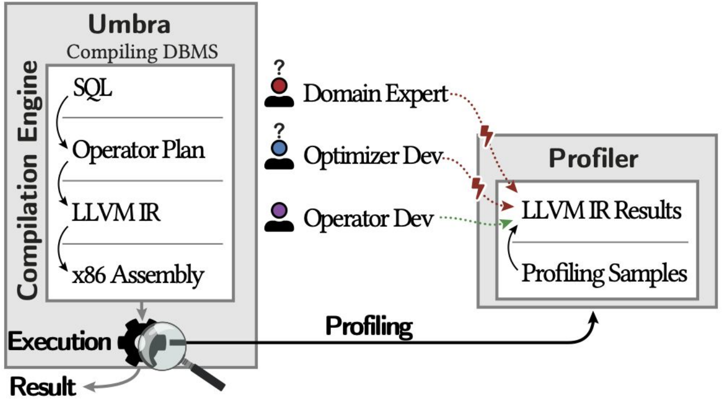
- Design and populate various databases for cellular coverage, and national parks that do not have food options etc
- Make an app that allows a user to plan a trip based on the data that you populated

#6 Visualizing Performance Profiling

Compiling DBMSs use multiple abstraction levels. However, traditional profilers only report the results at the lowest level.

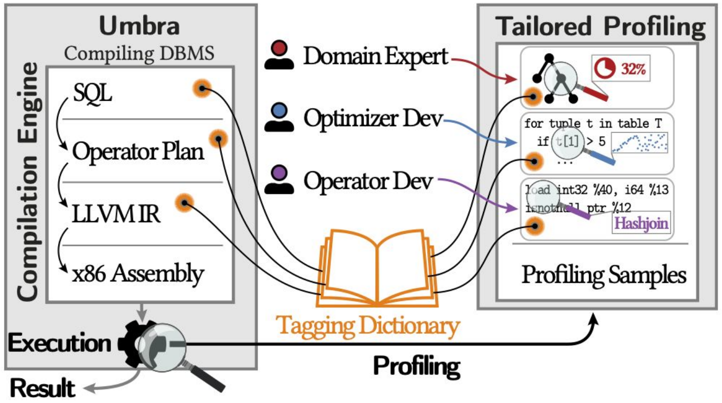
We developed a profiler that provides results at all abstraction levels.

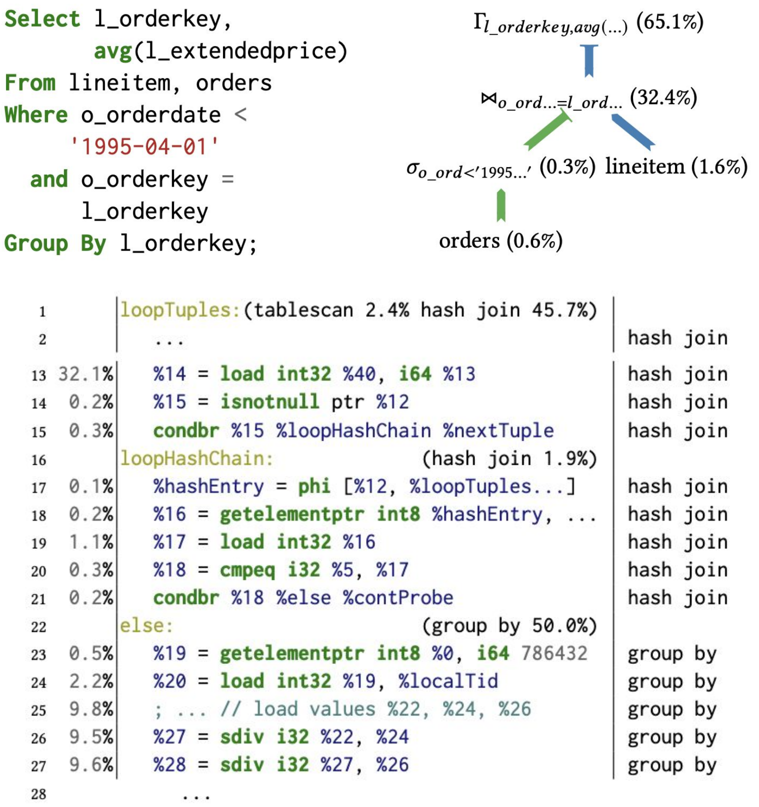
The profiler provides the data as formatted CSV-file, but the visualizations are still prototypes.



#6 Visualizing Performance Profiling

Examples:





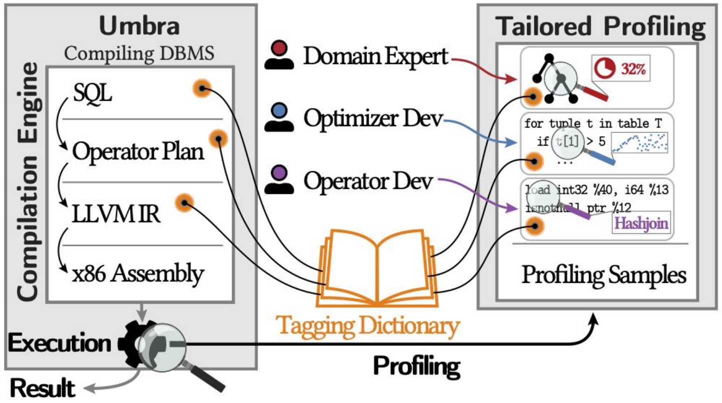
```
Select l_orderkey,
       avg(l_extendedprice)
From lineitem, orders
Where o_orderdate <
      '1995-04-01'
  and o_orderkey =
      l_orderkey
Group By l_orderkey;
```

$\Gamma_{l\_orderkey,avg(...)}$ (65.1%)
$\bowtie_{o\_ord...=l\_ord...}$ (32.4%)
$\sigma_{o\_ord<'1995...'}$ (0.3%)
lineitem (1.6%)
orders (0.6%)

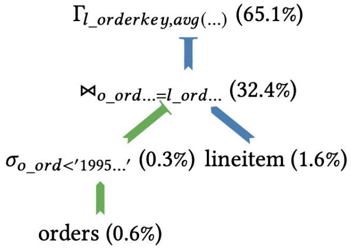
```
      loopTuples:(tablescan 2.4% hash join 45.7%)
         ...                                       hash join
32.1%    %14 = load int32 %40, i64 %13             hash join
 0.2%    %15 = isnotnull ptr %12                   hash join
 0.3%    condbr %15 %loopHashChain %nextTuple      hash join
      loopHashChain:              (hash join 1.9%)
 0.1%    %hashEntry = phi [%12, %loopTuples...]    hash join
 0.2%    %16 = getelementptr int8 %hashEntry, ...  hash join
 1.1%    %17 = load int32 %16                      hash join
 0.3%    %18 = cmpeq i32 %5, %17                   hash join
 0.2%    condbr %18 %else %contProbe               hash join
      else:                       (group by 50.0%)
 0.5%    %19 = getelementptr int8 %0, i64 786432   group by
 2.2%    %20 = load int32 %19, %localTid           group by
 9.8%    ; ... // load values %22, %24, %26        group by
 9.5%    %27 = sdiv i32 %22, %24                   group by
 9.6%    %28 = sdiv i32 %27, %26                   group by
           ...
```

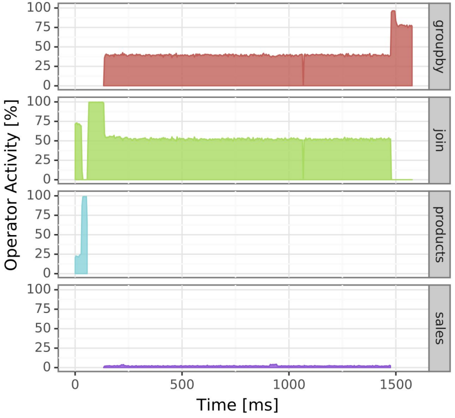
#6 Visualizing Performance Profiling

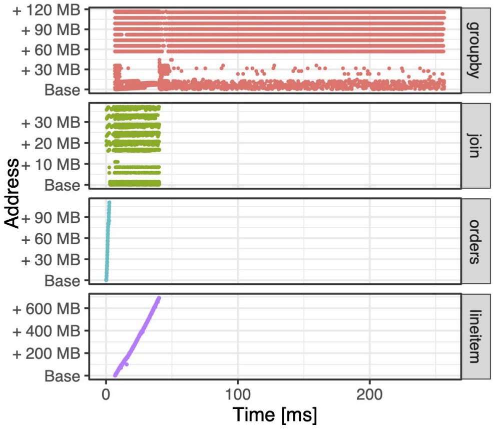
Examples:



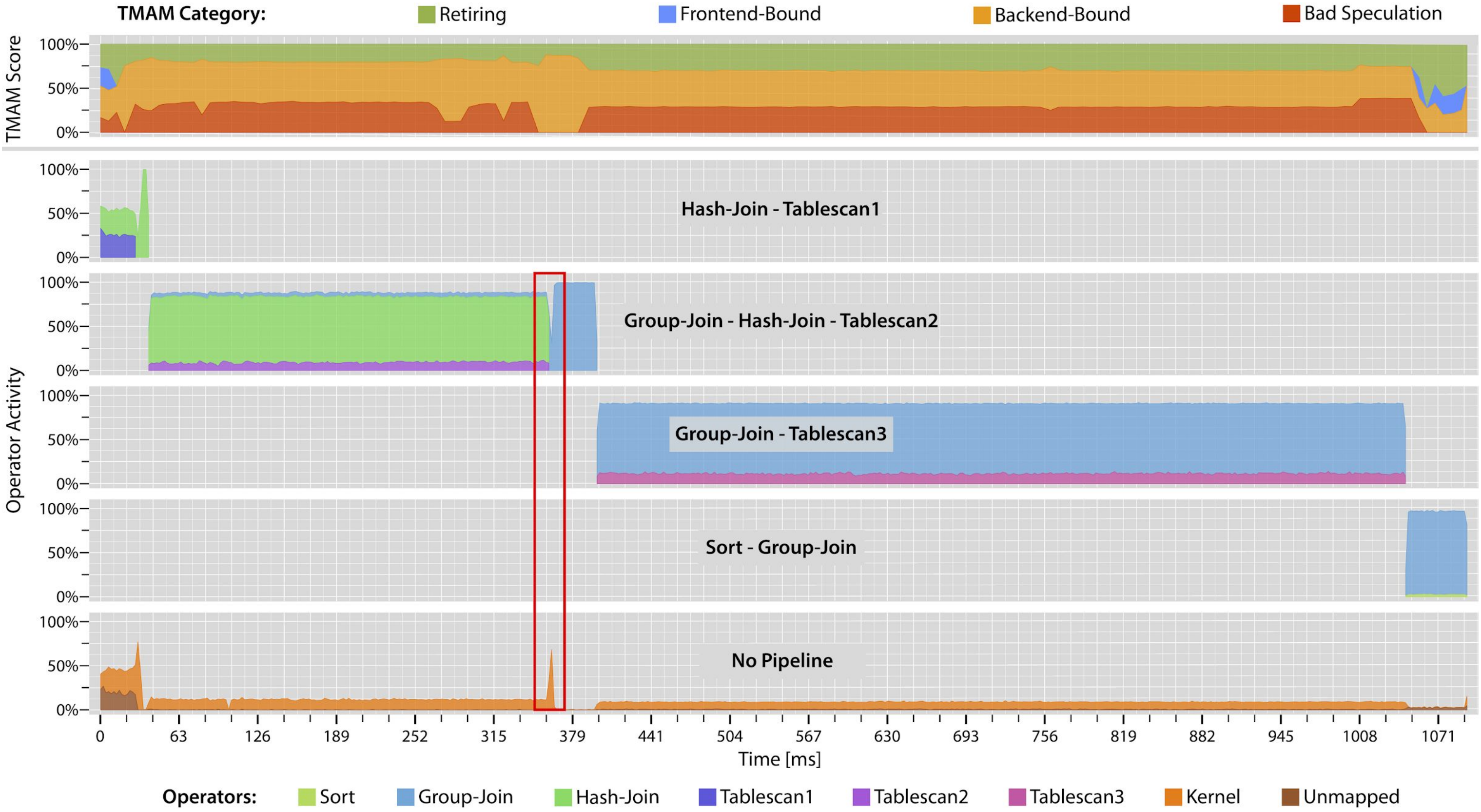
```
Select l_orderkey,
       avg(l_extendedprice)
From lineitem, orders
Where o_orderdate <
        '1995-04-01'
  and o_orderkey =
        l_orderkey
Group By l_orderkey;
```


$\Gamma_{l\_orderkey,avg(...)}$ (65.1%)
$\bowtie_{o\_ord...=l\_ord...}$ (32.4%)
$\sigma_{o\_ord<'1995...'}$ (0.3%)
lineitem (1.6%)
orders (0.6%)





#6 Visualizing Performance Profiling



#6 Visualizing Performance Profiling

Your tasks:

- Design suited visualizations for the results of the different abstraction levels.
- Build an app to report the results from the CSV-file for the different levels.

Will be definitely used by the chair!

operator	pipeline	uir_code	src_line	dump_ linenr	time	addr	phys_addr	ev_name	period	cpu
groupby1	groupby_join_join_tablescan_partsupp 139711406319616	%13228 = call i64 TextRuntime::hashCRC (%12311, %13168)	TextRuntime.cpp: 585	730	9151245228 344180	139698211034120	11286295560	mem_inst_retired. all_stores	500	4294967295
join2	join_tablescan_part 139998635408536	store int32 %8863, %8997	dump9.uir:496	496	9151245172 672389	139698115714648	9748225624	mem_inst_retired. all_stores	500	4294967295
sort1	sort_groupby 139998635275408	%289 = load data128 %left	dump9.uir:68	68	9151245727 020600	139698219426512	6695150288	mem_inst_retired. all_stores	500	4294967295

#7 Improving the Housing Market

ImmoScout recently started an API to their listings

- Find out what you can (legally) do with the data (ToS, request API key)
- Automatically extract good deals and notify the user
- Find dubious listings
- Visualize current trends by area/flat size

Also possible with other platforms (immowelt/wg-gesucht/ebay kleinanzeigen etc.)

A bit of inspiration...

SpiegelMining (http://www.dkriesel.com/spiegelmining)

BahnMining (http://www.dkriesel.com/blog/2019/1229_video_und_folien_meines_36c3-vortrags_bahnmining)

Talks also discuss ethics and code of conduct of mining public apis.
Check them out!

One More Thing ...

Look for open data sets !

Google dataset explorer: https://www.google.com/publicdata/directory

Amazon co-purchasing set: https://snap.stanford.edu/data/com-Amazon.html

Flights: http://stat-computing.org/dataexpo/2009/the-data.html

IMDB: https://www.imdb.com/interfaces/

Wiki: http://dumps.wikimedia.org