KIT
Karlsruher Institut für Technologie

Introduction to Otree and the Hidden Profile Task

About the Experiment

Adaptive Systems for Goalsetting in virtual Teams

- Objective:

https://de.cleanpng.com/png-4aq0qi/

Does applying the WOOP method in virtual meetings improve team outcomes in a collaborative task?

Does a graphical adaptation visualizing team members' goals in virtual meetings, result in improved team outcomes in a collaborative task?

Adaptive Systems for Goalsetting in virtual Teams

- Task: Decide for a innovative projects as team:

(1) *Virtual Reality Fitness Adventure Game*

(2) *AI-Powered Personalized Shopping Assistant*

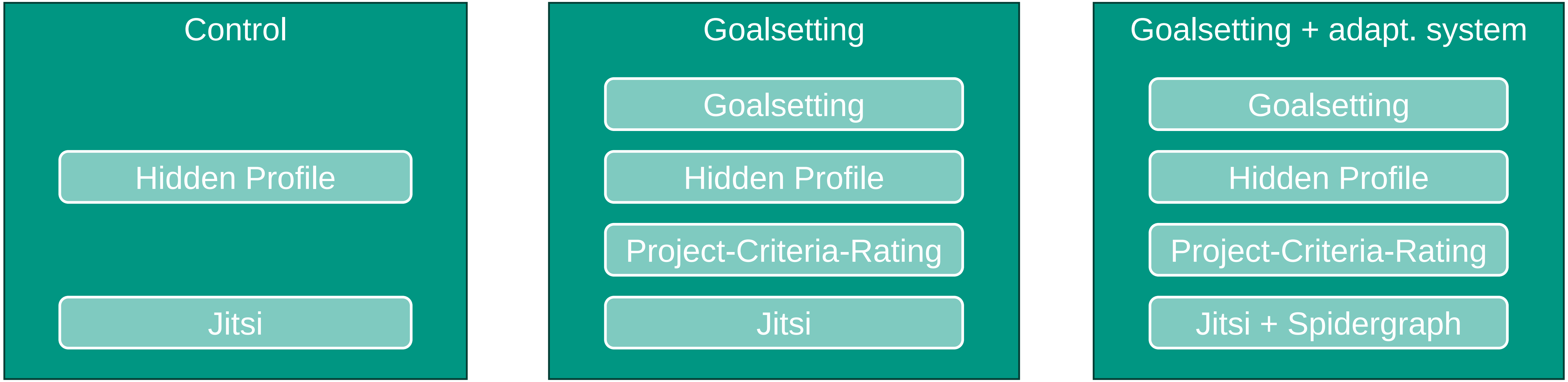
(3) *Smart Home Energy Management System*

- hidden profil = asynchron information

https://de.cleanpng.com/png-51e5cn/

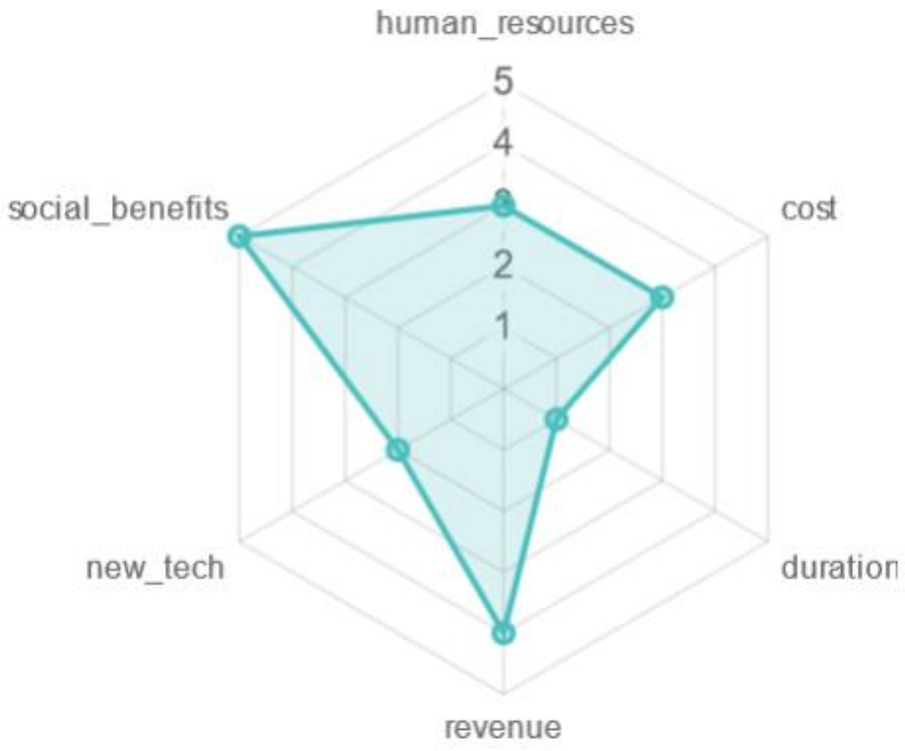
Adaptive Systems for Goalsetting in virtual Teams

- implementation in oTree
- use of Jitsi as open source video conference system
- 20 teams of four participant for each treatment in KD2Lab:



Control	Goalsetting	Goalsetting + adapt. system
	Goalsetting	Goalsetting
Hidden Profile	Hidden Profile	Hidden Profile
	Project-Criteria-Rating	Project-Criteria-Rating
Jitsi	Jitsi	Jitsi + Spidergraph

Adaptive Systems for Goalsetting in virtual Teams

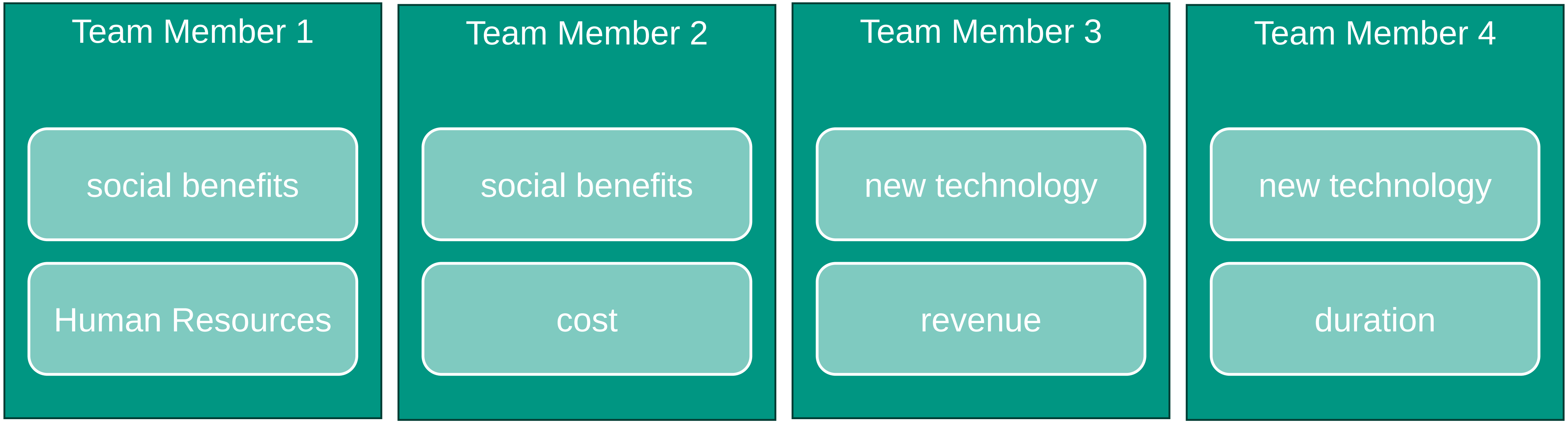


eigene Darstellung

- six possible goals, that innovative projects should fulfill, are provided
- individual selection of the most important goal
- individual allocation of 18 points to the six given goals (1 to 5 points per goal)
- individual mental contrasting to the Wish (= most important goal) with WOOP

Adaptive Systems for Goalsetting in virtual Teams

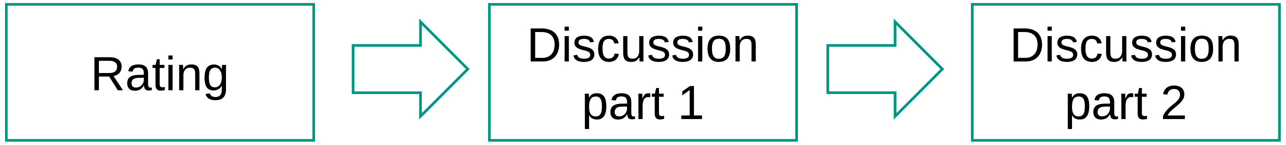
- allocation of information in hidden profile task:



Team Member 1	Team Member 2	Team Member 3	Team Member 4
social benefits	social benefits	new technology	new technology
Human Resources	cost	revenue	duration

Adaptive Systems for Goalsetting in virtual Teams

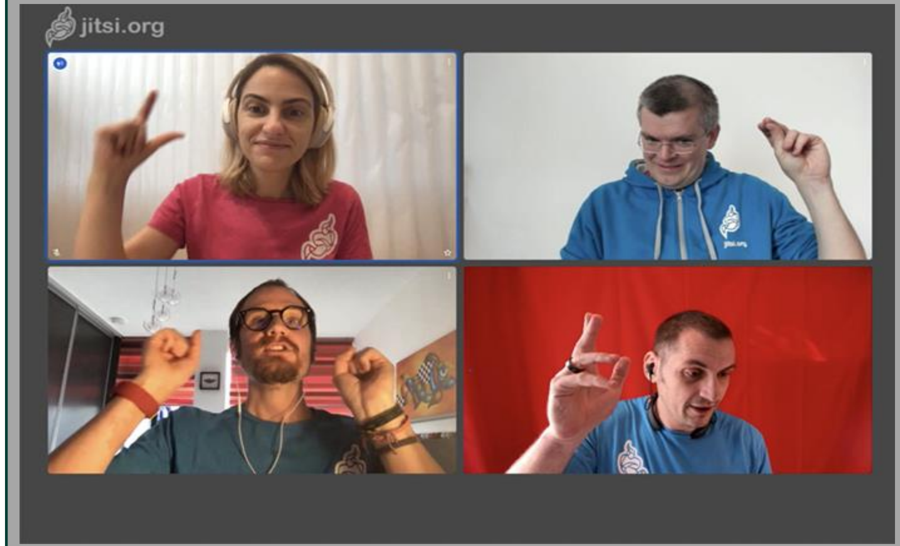
- choice/discussion: which project fulfills which criteria best
- discussion: which project to choose as a team
- dependent Variables:
  - has the team reached consensus?
  - and if so, how long did this take?


Rating

Discussion part 1

Discussion part 2

“Adaptive“ Video Meeting System



<u>Side Panel</u>

Spidergraph with Team Goals

(average of individual goals)

https://jitsi.org/blog/e2ee/

Introduction

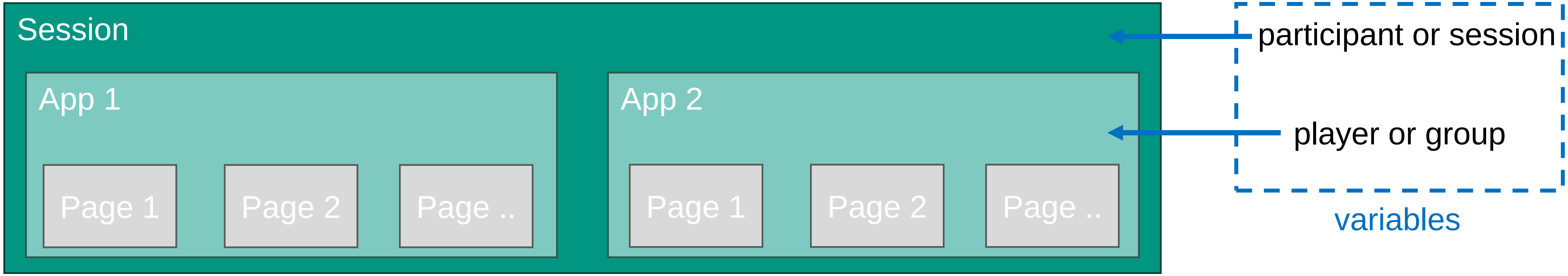
General Information

- Use of oTree 5 (not 3) – be careful if working with ChatGPT
- oTree Documentation https://otree.readthedocs.io/en/latest/
- Meetup: Once in 2 weeks.

- GitHub: https://github.com/AnujaHari87/hapshiddenprofile/tree/dev_new

- First Tasks:
  - install oTree (pip install otree)
  - hands on: create your first app (simple survey with questions of your choice)
    prompt → „otree startproject my_first_project“; „otree startapp my_app“
  - test your app on localhost: prompt → „otree devserver“
- Next Steps:
  - clone Repository
  - create new branch

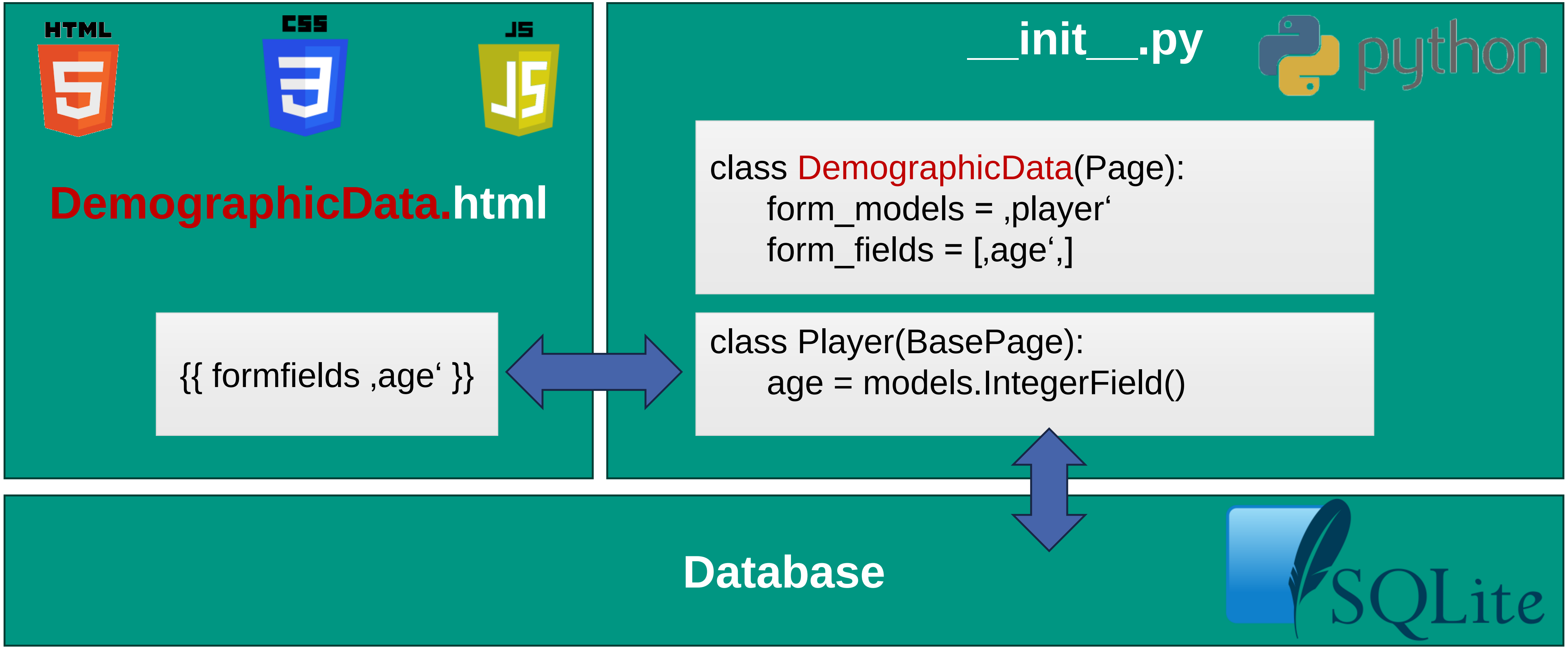
Otree Structure



- An app is part of an session and consists of multiple pages
- at app level there are player and group variables, at session level there are participant and session variables
- You could access participant and session data at app level: e.g. player.session.name_of_var
- an app consists of an __init__.py and some html-files
- the python-part of an app is mostly done in the __init__.py
- the session is organized in the setting.py

Otree Structure

- in oTree 5 the most of the implementation is done inside the __init__.py



```
{{ formfields ‚age‘ }}
```

```
class DemographicData(Page):
    form_models = ‚player‘
    form_fields = [‚age‘,]
```

```
class Player(BasePage):
    age = models.IntegerField()
```

__init__.py

- Other classes:
  - C(BaseConstants) → define constants for the actual app
  - Subsession(BaseSubsession)
  - Group(BaseGroup) → define variables on group-level
  - SomeWaitPage(WaitPage) → Page waits until all players have arrived

- Other Formmodels:
  - models.CurrencyField()
  - models.StringField()
  - models.BooleanField()
  - ..

- defined inside class Player or class Group
- used to store Variables for Database
- accessible in template via formfields

Methodes to control experiment flow

- inside of Page-Class (in __init__.py)
- has to be @staticmethode (in oTree5)

- def get_form_fields(player: Player)
- def before_next_page(player, timeout_happened)
- def vars_for_template(player)
- def js_vars(player)
- def live_method(player, data)

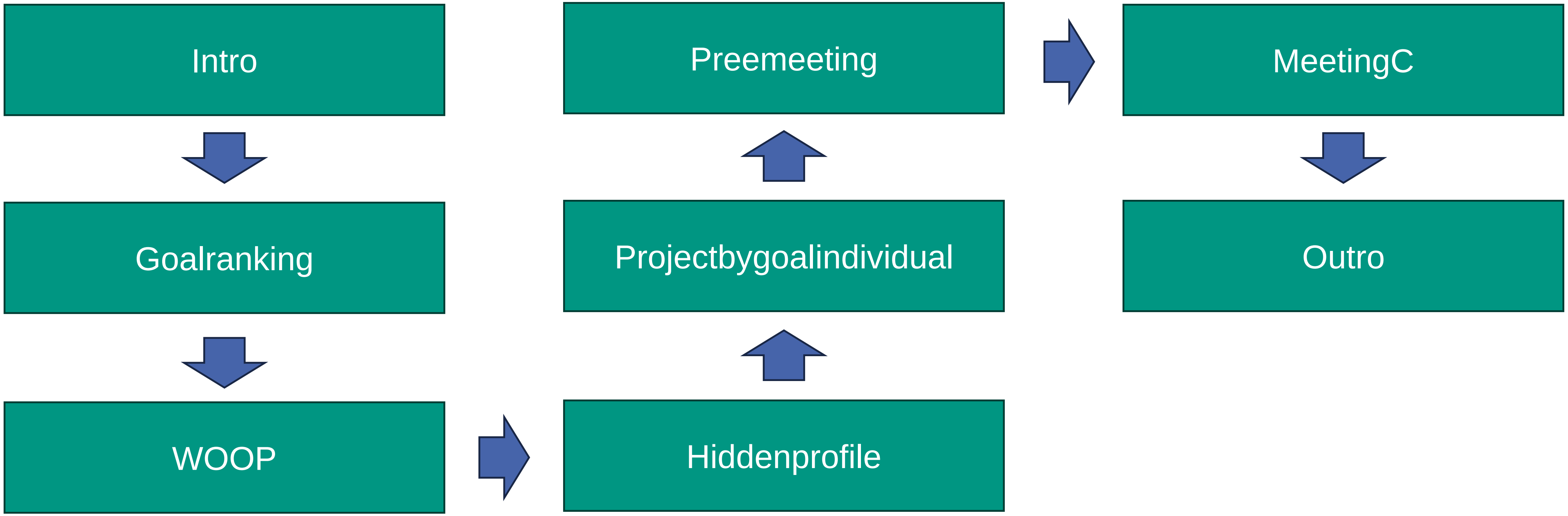
setting.py

- SESSION_CONFIGS = [

```
dict( name='Pilot‘,
        num_demo_participants=4,
        app_sequence=[‚DemographicData‘]
),
]
```

- PARTICIPANT_FIELDS = [ ]
- SESSION_FIELDS = [ ]

Apps and Sessions in the hidden profile task



Intro

Goalranking

WOOP

Hiddenprofile

Projectbygoalindividual

Preemeeting

MeetingC

Outro

Open Topics

- Improve Look & Feel
- Control experiment-flow
  (e.g. do not continue with the experiment until certain requirements are met)
- Participate in pilot testing in the lab
- Gather feedback on game understanding, usability & implement changes
  Implement pre- and post- questionnaires
- Finalize instruction wording
- Check data quality, format

Optional Topics:

- Implement a first version of Audio share element
  - Speech share using Jitsi API
- Implement a first version of a cognitive workload detection element
  - OpenCV → module for eye blinking in videos (live or offline?)