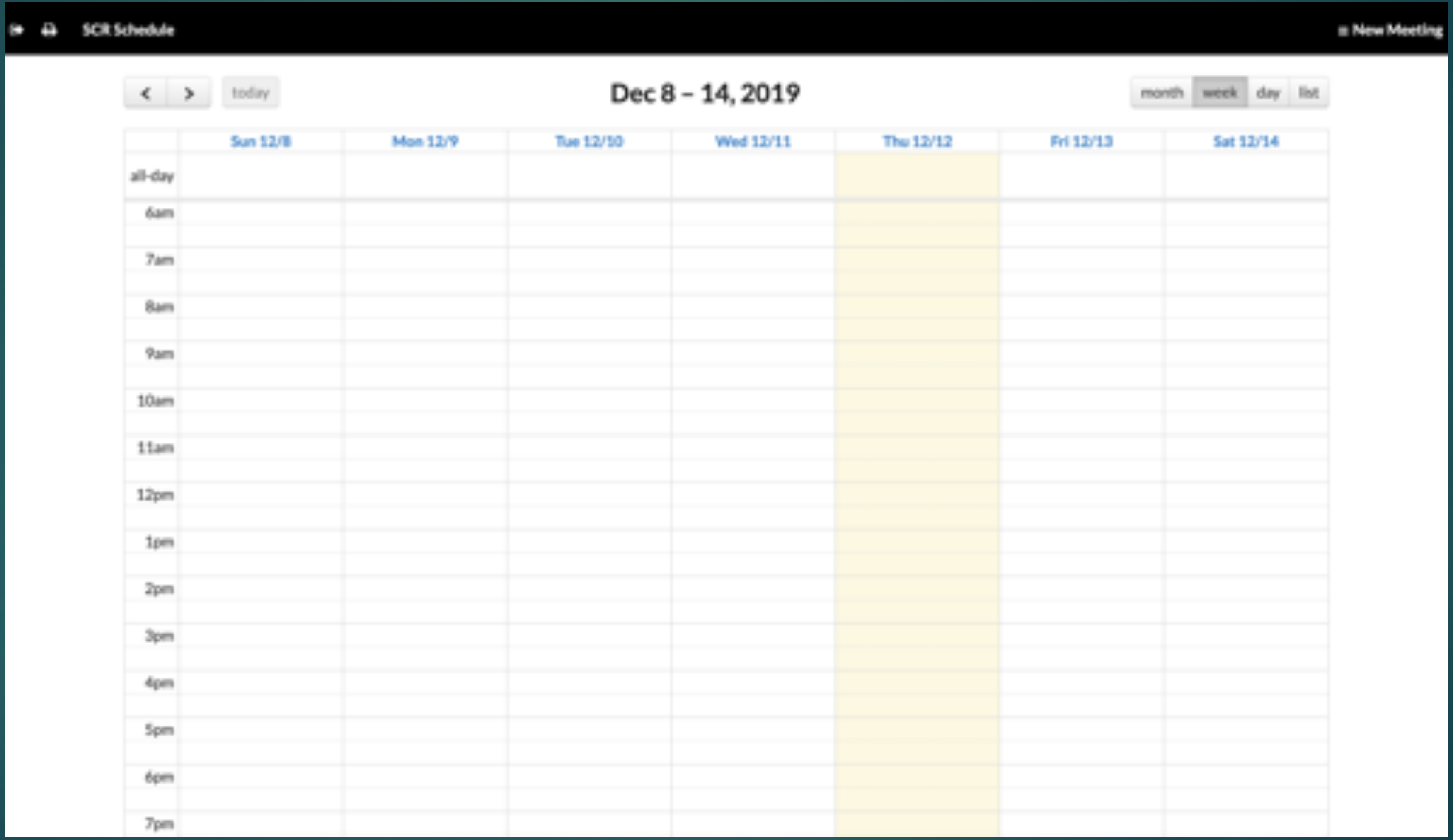
# Fall 2019 Demo/Walkthrough

A SUMMARY OF HOW EVERYTHING OPERATES WHEN LIVE

# Part 1: Meetings



Dec 8 – 14, 2019

| | Sun 12/8 | Mon 12/9 | Tue 12/10 | Wed 12/11 | Thu 12/12 | Fri 12/13 | Sat 12/14 |
|---|---|---|---|---|---|---|---|
| all-day | | | | | | | |
| 6am | | | | | | | |
| 7am | | | | | | | |
| 8am | | | | | | | |
| 9am | | | | | | | |
| 10am | | | | | | | |
| 11am | | | | | | | |
| 12pm | | | | | | | |
| 1pm | | | | | | | |
| 2pm | | | | | | | |
| 3pm | | | | | | | |
| 4pm | | | | | | | |
| 5pm | | | | | | | |
| 6pm | | | | | | | |
| 7pm | | | | | | | |

# Website

- **http://128.113.122.178**
- Option to view the current Smart Conference Room Schedule as a guest
- User enters the meeting's date, time, attendees' email, etc. From here, the information gets sent to the Preference Server.
  - Alternatively, we can add meetings through LESA's Google Calendar with the following commands under arunas on the Linux Machine
    - cd Documents/GoogleCalendar/GoogleCalendarIntegration
    - source calendar/bin/activate
    - python googlecalendar_scrschedule.py

# Preference Server/HVAC Server

- The Preference Server will look up the information of the attendees (and/or create accounts for new attendees/users).
- The Preference Server will communicate with the HVAC Server and set the room’s temperature to the average of these temperatures for the upcoming meeting.
  - As discussed for this demo, the temperature is set 10 minutes before the meeting begins.
- At the end of a meeting, unless there is a meeting coming up immediately after, the room’s temperature is set back to the room’s Building Management System.

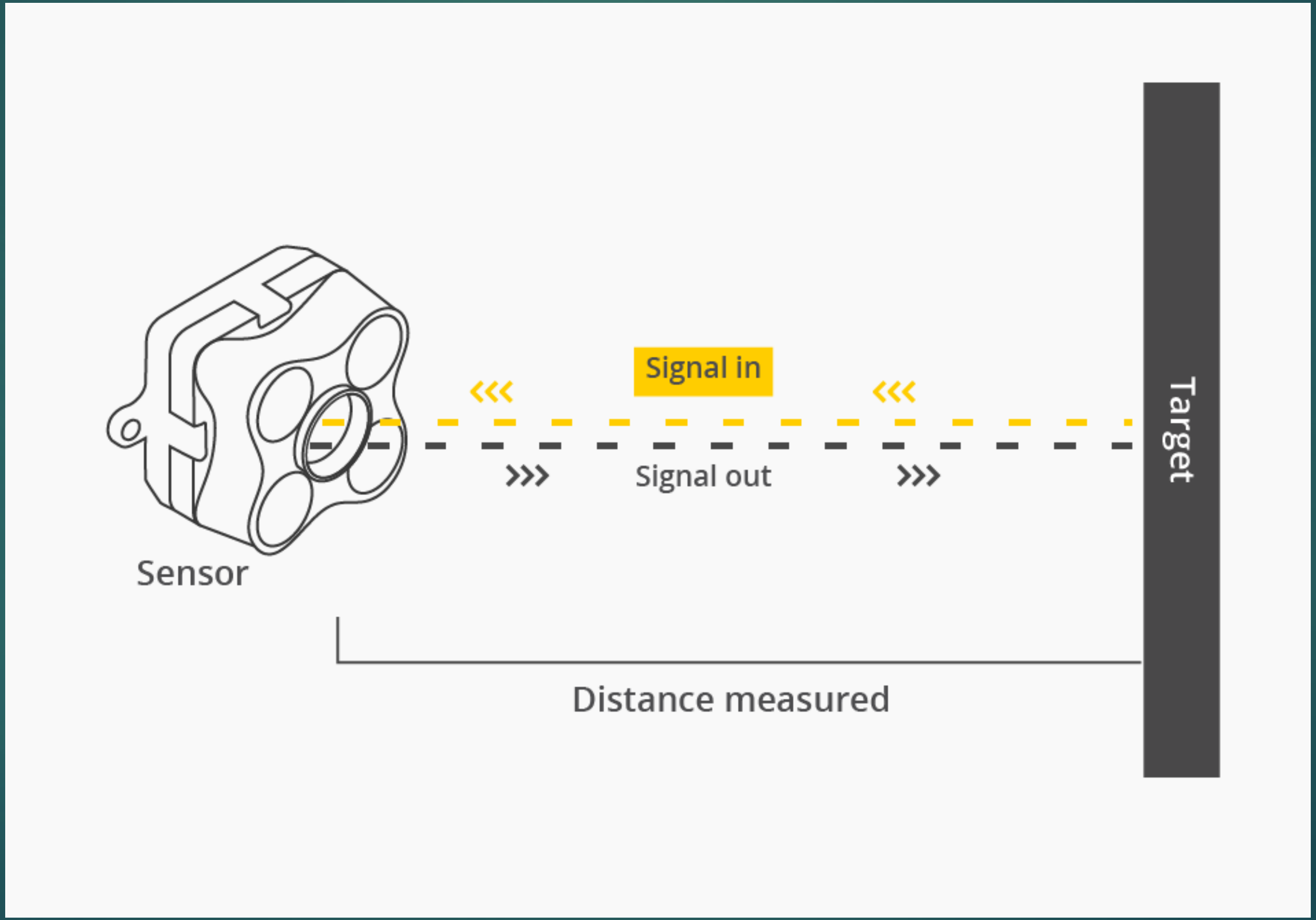
# This Meeting

- Verification of pre-heat/pre-cooling: I set both of your temperature preferences to something higher than the original temperature, because it is much easier to heat the room than to cool the room in 10 minutes .

- Set temperature today at 9:50 am (Ten minutes before our meeting).
  - We can see the current change in temperature
    - **cat HVAC/Month.Day.Year.txt** (in this case: cat HVAC/12.13.2019.txt)

- The temperature has changed to the temperature **currently on the screen**, in the past 10 minutes

# Part 2: Tracking and Tagging



# Tracking

- The foundation of the tracking work originates from TK and Arunas's work from a prior LESA Industry-Academia Day
  - The Time of Flight sensors on the ceiling are used to track occupants in the SCR
- I have **output** to my terminal window the tracking.
  - Each list represents an occupant
    - [x position, y position, height, current position, previous x position, previous y position, previous position, label]
      - Position: Sitting or standing
      - Label: Used to reference occupants for tagging
- Due to the foundational TOF code being worked on by others for the upcoming demo, there are still some issues that may arise
  - Roaming around the entrance area, may cause the sensors to register a new occupant in the SCR
    - At the prior demo, they put tape on the floor to guide the guests around to avoid this issue

# Logging Onto the Preference Client

- The preference clients require an email address to login.
    - This change allows a secretary to use the already-known email addresses of attendees who requested a meeting over email
        - There is no need to know custom usernames of attendees
- People who were not originally on the list can now login to the preference clients as a guest, or create an account like before
- From here, we can login to the preference clients (I have already added your RPI email)

# Tagging

- As in the past, once logged in, the preference client receives their profile menu from the Preference Server
- But additionally, the user’s information is sent to a tracking coordinator server, where the occupant who has logged on is now tagged.
  - To verify tagging now works, I have printed some output onto one of my terminal windows.
- Technically, the person tagged ends up being the occupant who has the closest distance to the preference client, which was just logged in from.

# Part 3: What's next

- 1. Implementing the algorithm that finds the time to begin pre-heating/pre-cooling
  - Once the algorithm is created, this should be very easy and quick to implement
    - The only that would need to be changed is a variable in the Preference Server, that declares how many seconds before the meeting, should I set the temperature
- 2. Implementing the algorithm that will satisfy the temperatures of occupants
- 3. Anything related to the Industry Days Demo in April

# Relevant Code

- HVAC-and-Lighting-Preference-System Repository
- SCR Schedule Website Repository
- Miscellaneous-SCR-Projects Repository
- Not my code, but dependent on the code above
  - Toufiq’s HVAC Server Code (192.168.0.36)
  - Martin’s scr.scr_control (Private repository, which can be found in the LESA Organization GitHub Page)