NTNU

Kunnskap for en bedre verden

Course Project Introduction

Computer communication and network programming

Outline

1. Course content and examination
2. The case: smart greenhouse
3. Requirements
4. Non-functional requirements
5. Grading

Outline

1. **Course content and examination**
2. Scenario: smart greenhouse
3. Requirements
4. Non-functional requirements
5. Grading

Course content

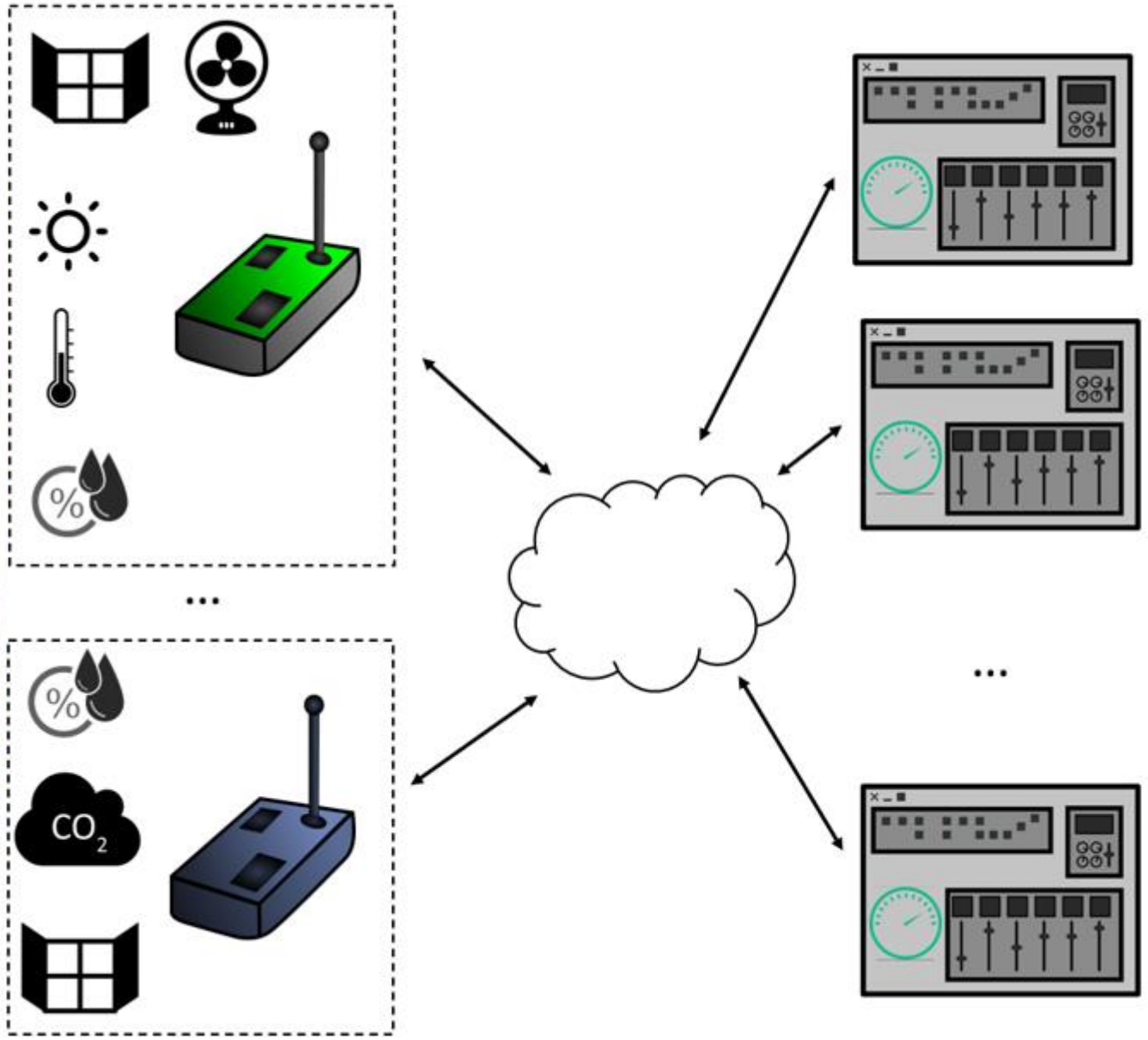
- Network protocol theory: individual written exam = 40% of grade
- Network programming: group project = 60% of grade
- Final grade calculation:

	Written exam (column)					
Project (row)	**A**	**B**	**C**	**D**	**E**	**F**
A	A	A	B	B	C	F
B	B	B	B	C	C	F
C	B	C	C	C	D	F
D	C	C	D	D	D	F
E	C	D	D	E	E	F
F	F	F	F	F	F	F

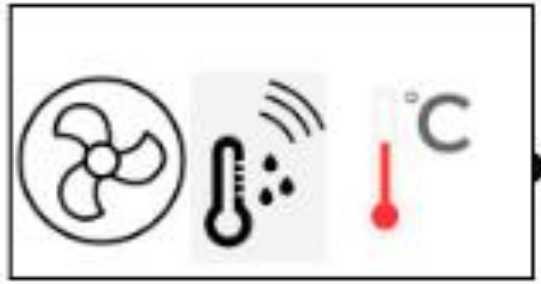
Outline

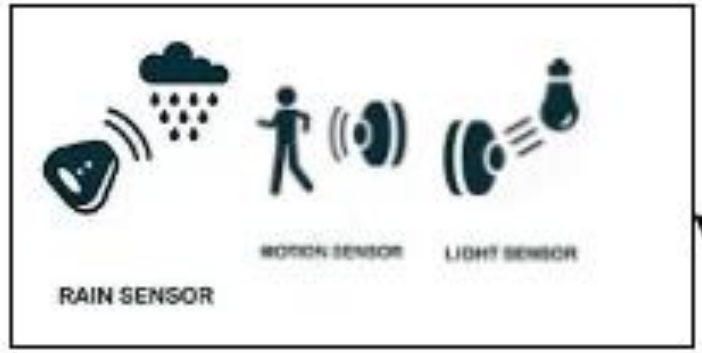
1. Course content and examination
2. **Scenario: smart greenhouse**
3. Requirements
4. Non-functional requirements
5. Grading

Smart greenhouse

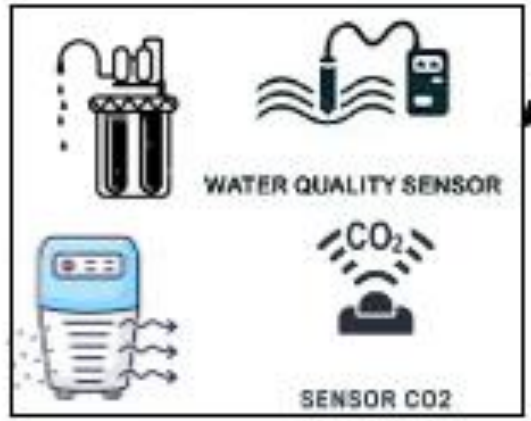


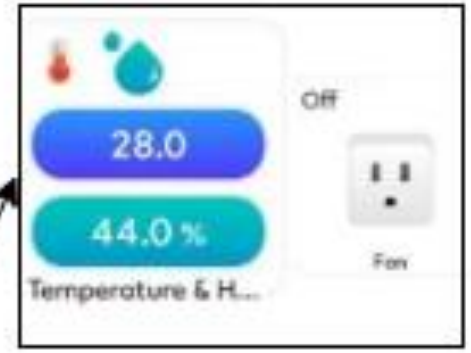
GREEN HOUSE

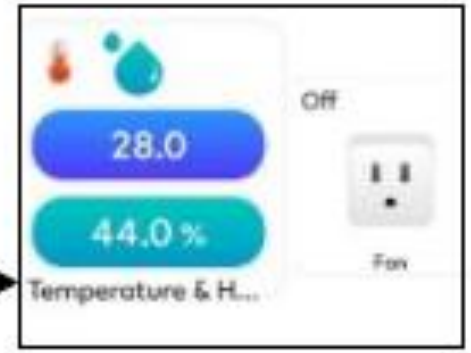

°C

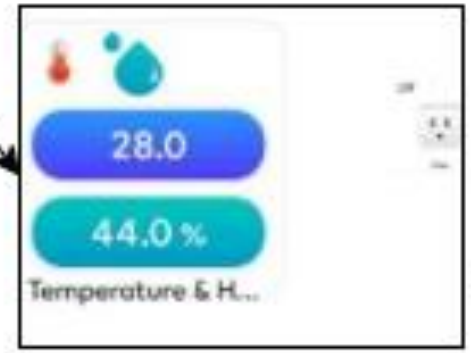

RAIN SENSOR
MOTION SENSOR
LIGHT SENSOR


HUMIDITY SENSOR
FLAME DETECTOR


WATER QUALITY SENSOR
CO2
SENSOR CO2

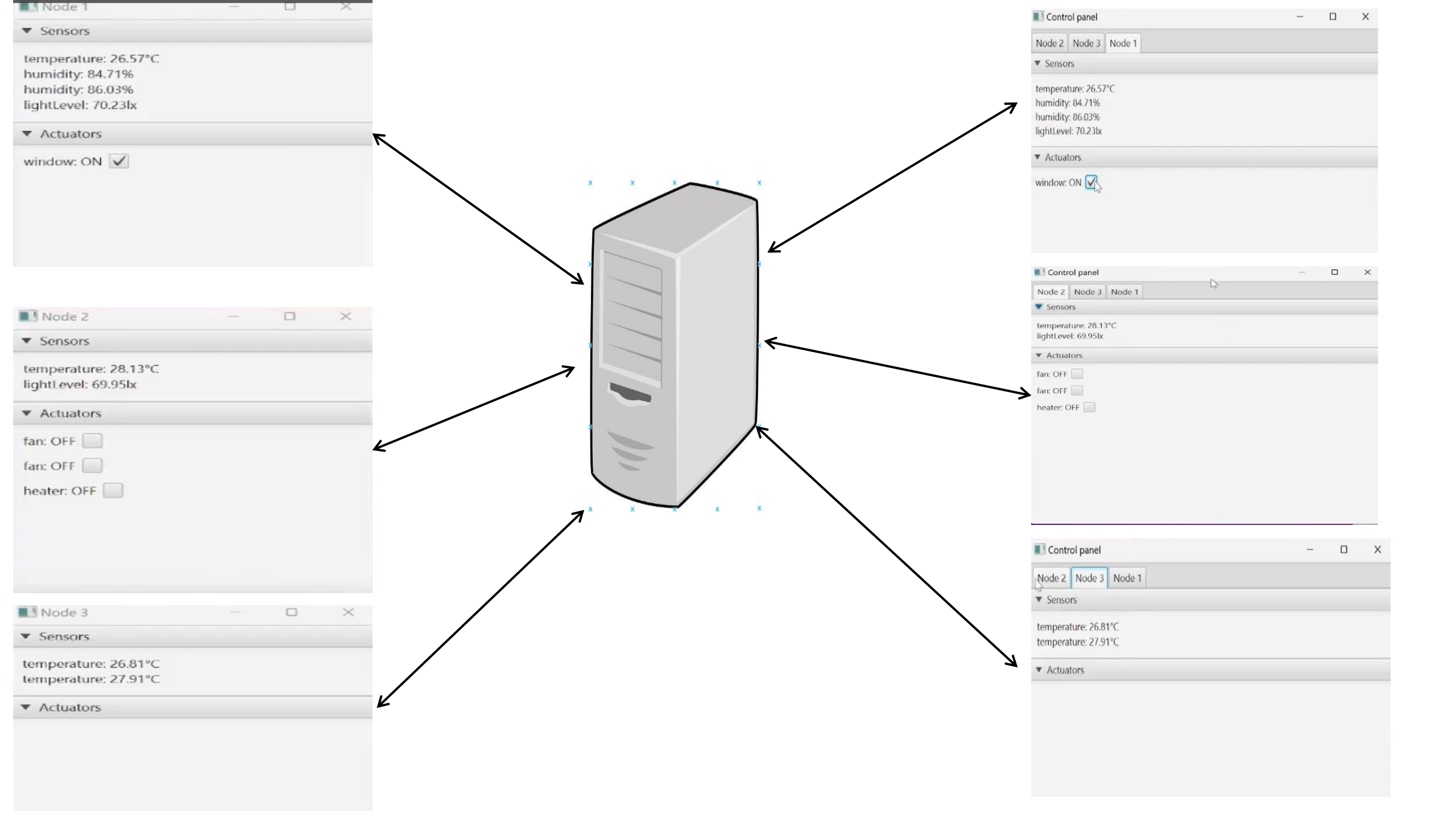

28.0
44.0 %
Off
Fan
Temperature & H...


28.0
44.0 %
Off
Fan
Temperature & H...


28.0
44.0 %
Temperature & H...

Simulation

- No physical sensor nodes
- Simulate sensor nodes (and a greenhouse) as JavaFX application(s) (use your programming-2 skill)


Node 1
Sensors
temperature: 26.57°C
humidity: 84.71%
humidity: 86.03%
lightLevel: 70.23lx
Actuators
window: ON
Node 2
Sensors
temperature: 28.13°C
lightLevel: 69.95lx
Actuators
fan: OFF
fan: OFF
heater: OFF
Node 3
Sensors
temperature: 26.81°C
temperature: 27.91°C
Actuators
Control panel
Node 2
Node 3
Node 1
Sensors
temperature: 26.57°C
humidity: 84.71%
humidity: 86.03%
lightLevel: 70.23lx
Actuators
window: ON
Control panel
Node 2
Node 3
Node 1
Sensors
temperature: 28.13°C
lightLevel: 69.95lx
Actuators
fan: OFF
fan: OFF
heater: OFF
Control panel
Node 2
Node 3
Node 1
Sensors
temperature: 26.81°C
temperature: 27.91°C
Actuators

Outline

1. Course content and examination
2. Scenario: smart greenhouse
3. Requirements
4. Non-functional requirements
5. Grading

Outline

1. Course content and examination
2. The case: smart greenhouse
3. **Requirements**
4. Non-functional requirements
5. Grading

Requirements

- Use raw sockets: TCP or UDP. No MQTT, HTTP, etc.
- Use JavaFX as GUI.

Your tasks

1. Design application-layer communication protocol
2. Describe the protocol and its architecture
3. Implement the protocol:
    1. Sensor/actuator node
    2. Control panel node
    3. The server
4. Create a video-presentation

For details, see Blackboard > Learning Materials > Project > Project Requirements

Outline

1. Course content and examination
2. Scenario: smart greenhouse
3. Requirements
4. **Non-functional requirements**
5. Grading

Non-functional requirements

- Take care of code quality
  - Proper cohesion and coupling (classes and methods, dependencies)
  - JavaDoc for all public methods
- Use version control
- Use agile work methodology:
  - Work in iterations (sprints)
  - Document the sprints:
    - What were the goal(s)?
    - How where the tasks distributed?
    - What was achieved?
    - Your own reflection - what went good, what needs to be improved?

Outline

1. Course content and examination
2. Scenario: smart greenhouse
3. Template code
4. Requirements
5. Non-functional requirements
6. **Grading**

Grading

- All mandatory parts covered? ➔ Grade = C
- Something missing? ➔ D or E (or F?)
- Some extras implemented? Grade += 1 OR 2

Possible extras

- Something communication-related
- Examples:
  - Data encryption or secure sockets
  - Reconnection on missed connectivity
  - Unique addresses for sensor nodes
  - File/Image transmission
  - Data on different resolutions (ex., hourly average)
  - Related to QoS
  - Compression Techniques

More on grading

- See Requirements