SESE Projects 2020 (EES + MPSEES)

Projects Overview

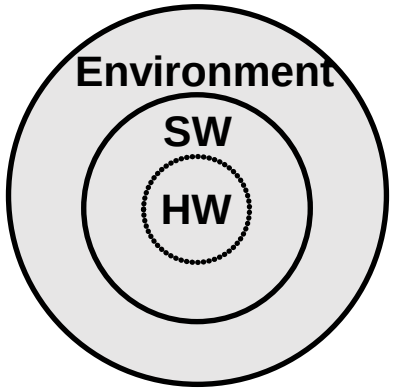
Sabine Glesner, Verena Klös, Paul Kogel,
Willie Szollmann

Technische Universität Berlin

SESE

A brief note on Embedded Systems

- Embedded systems **control** an **analogous** technical environment with **digital** components (hardware + software)
- Embedded systems **interact** with technical environment via **sensors and actuators**



A brief note on Embedded Systems (continued)

Characteristics:

- High degree of **concurrency**
- Timing important, often **real-time** requirements
- **Liveness** and **reactivity** (non-termination, events)
- High quality requirements
- Typically **networked**
- Cheap hardware
  - **Limited resources** (memory, energy etc.)
  - **Unreliable** sensors and actuators

Webots

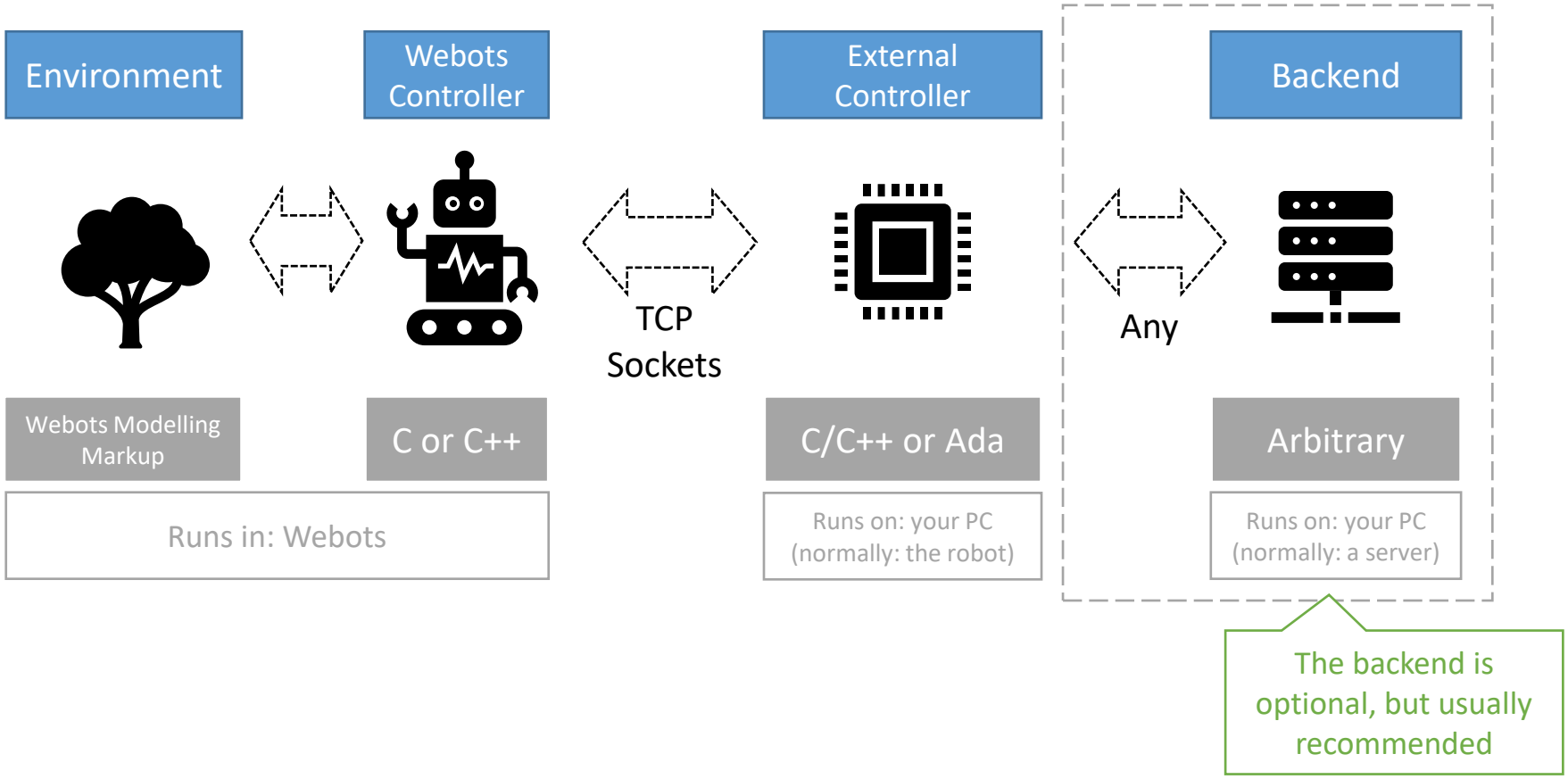
- Due to Covid-19: we are using **Webots virtual robots**!
- Download: https://cyberbotics.com/

- Webots allows you to build custom **environments** and **robots**
- Has rich library of **existing components**, including various sensors and actuators
- Provides ***Tinkerbots*** subset, which is especially beginner-friendly

- Each Webots robot has a ***controller*** that controls its sensors and actuators
- We are using a ***proxy architecture***, where the actual **controller** is **external** (see next slide)

General Architecture: Overview

Your project **must** use the following architecture:



General Architecture: Details

- The **Webots controller** is a **TCP socket server**:
  - Receives + executes commands from the external controller
  - Passes sensor data to the external controller
- Each Webots robot has its **own instance** of the **external controller**
- External controllers can communicate with each other. Webots controllers must not.
- The backend can communicate with all external controllers. It must not communicate with Webots controllers.

- More details on allowed programming languages in the individual **project descriptions**!

General Requirements

- Cope with unstable connections, communication delay and complete communication breakdowns
- Cope with “lost robots”
- Cope with unreliable sensor data
- The number of robots may change
- The environment may change

- Due to “virtual robots“: **simulate** environmental changes and communication faults

Additional Requirements (Master Students)

- **Quality assurance**:
  - Develop a QA concept
  - Define coding standards + clear interfaces
  - Test: test plans, unit tests, integration tests, ..
  - Verify: formal verification where possible + reasonable
- **Analysis and optimization**:
  - Optimization tasks
  - Analyze + optimize code where possible and reasonable

Topics

We offer three project topics:

Autonomous cab-on-demand service

Smart assembly line

Safe Reinforcement Learning

- Slides with detailed topic descriptions on **ISIS**!
- Our proposals are starting points. Extend them with your **own ideas**!

Useful Links

- **Webots User Guide**
  https://cyberbotics.com/doc/guide/

- **Webots Quickstart Tutorials**
  https://cyberbotics.com/doc/guide/tutorials

- **Tinkerbots User Guide**
  https://cyberbotics.com/doc/guide/tinkerbots

- **Interfacing Webots to Third Party Software with TCP/IP:**
  https://cyberbotics.com/doc/guide/interfacing-webots-to-third-party-software-with-tcp-ip