Securing Smart Environments with Authentic Execution

Candidate:
Gianluca Scopelliti

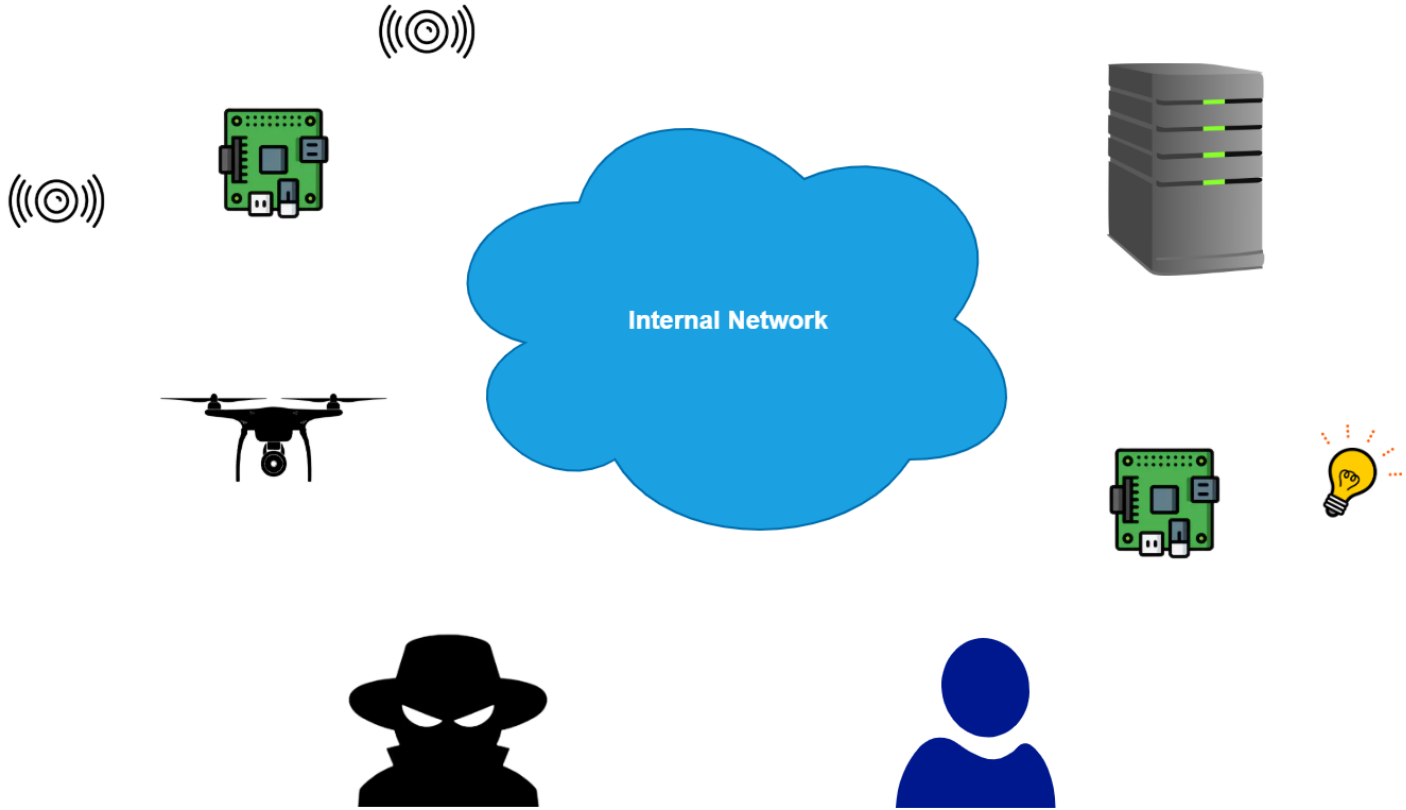
Thesis advisor:
Prof. Cataldo Basile

Supervisors:
Prof. Frank Piessens
Dr. Jan Tobias Mühlberg
Ir. Fritz Alder

What are Smart Environments?

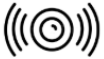
- **Distributed** applications
- **Connected** to the physical world
- **Heterogeneous**
- (optional) user intervention
- Examples:
  - Smart home
  - Smart farming

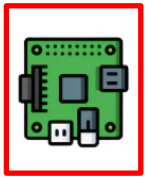
Security Issues

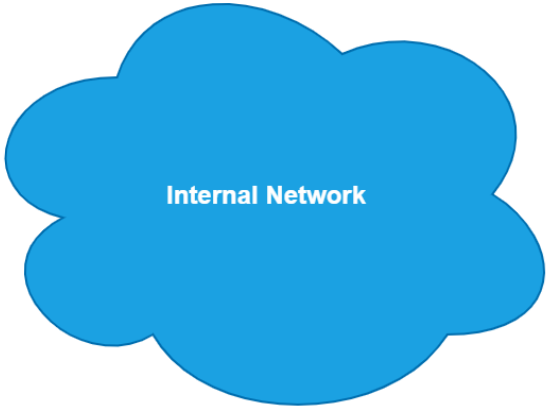


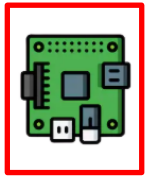
Security Issues

Authenticity of the software running on each node



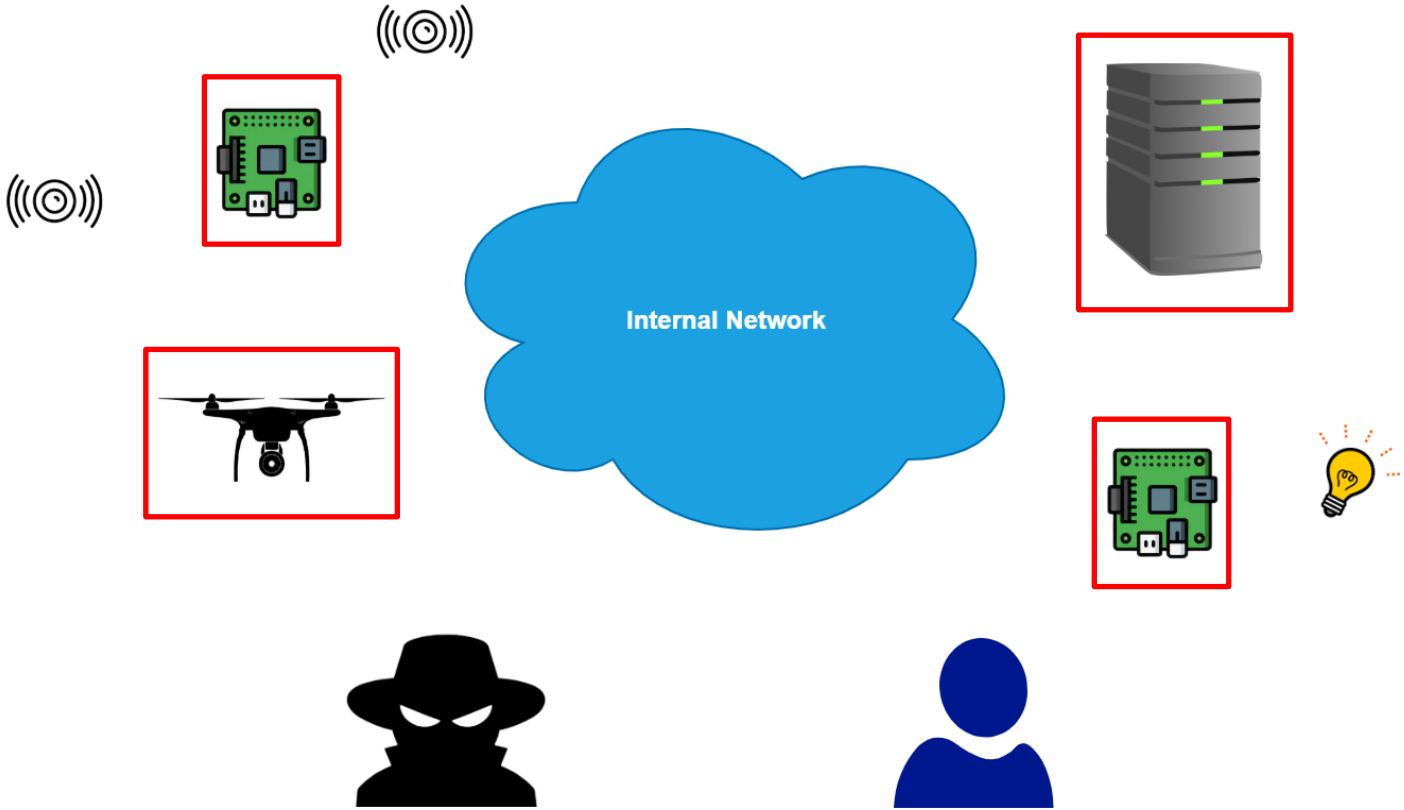






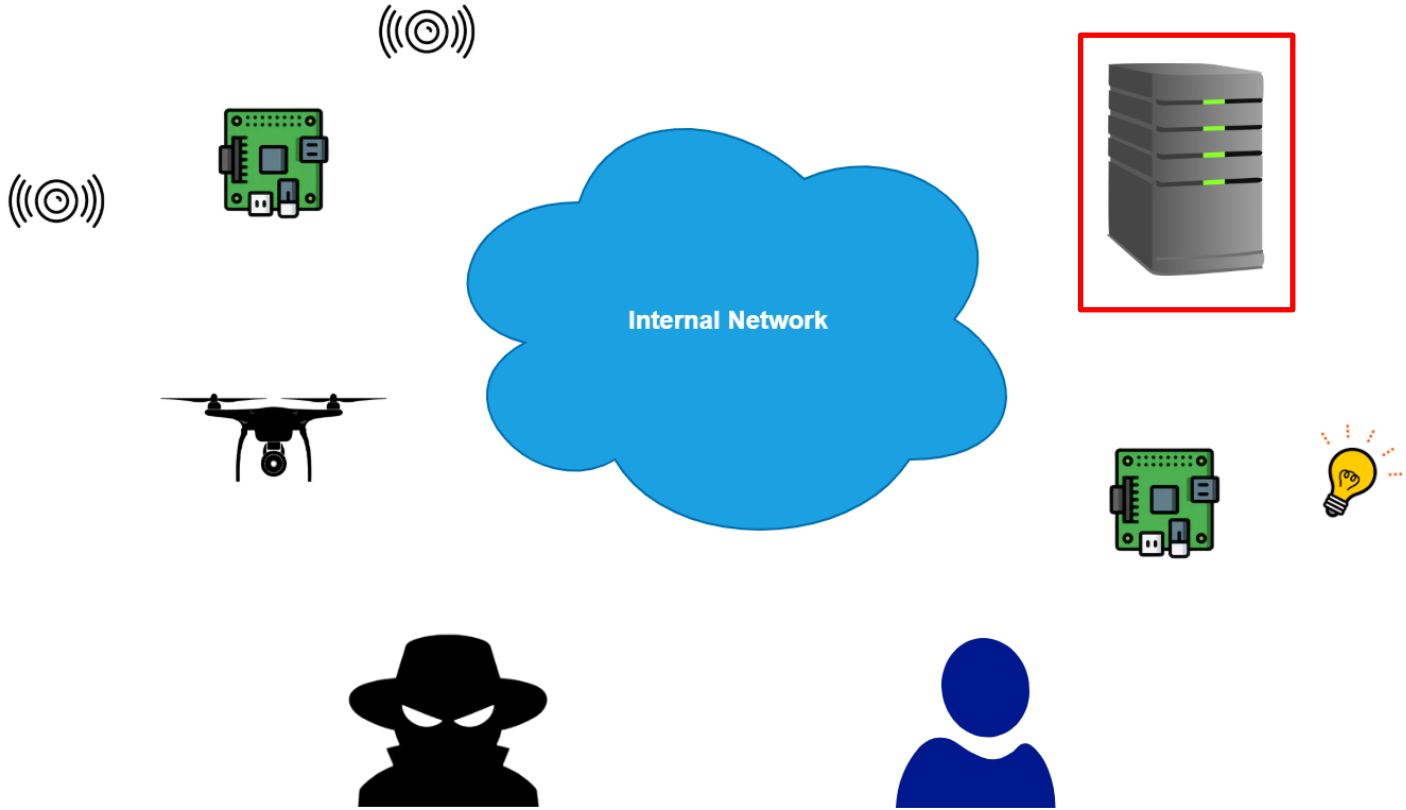
Security Issues

Confidentiality & integrity of sensitive data located in memory



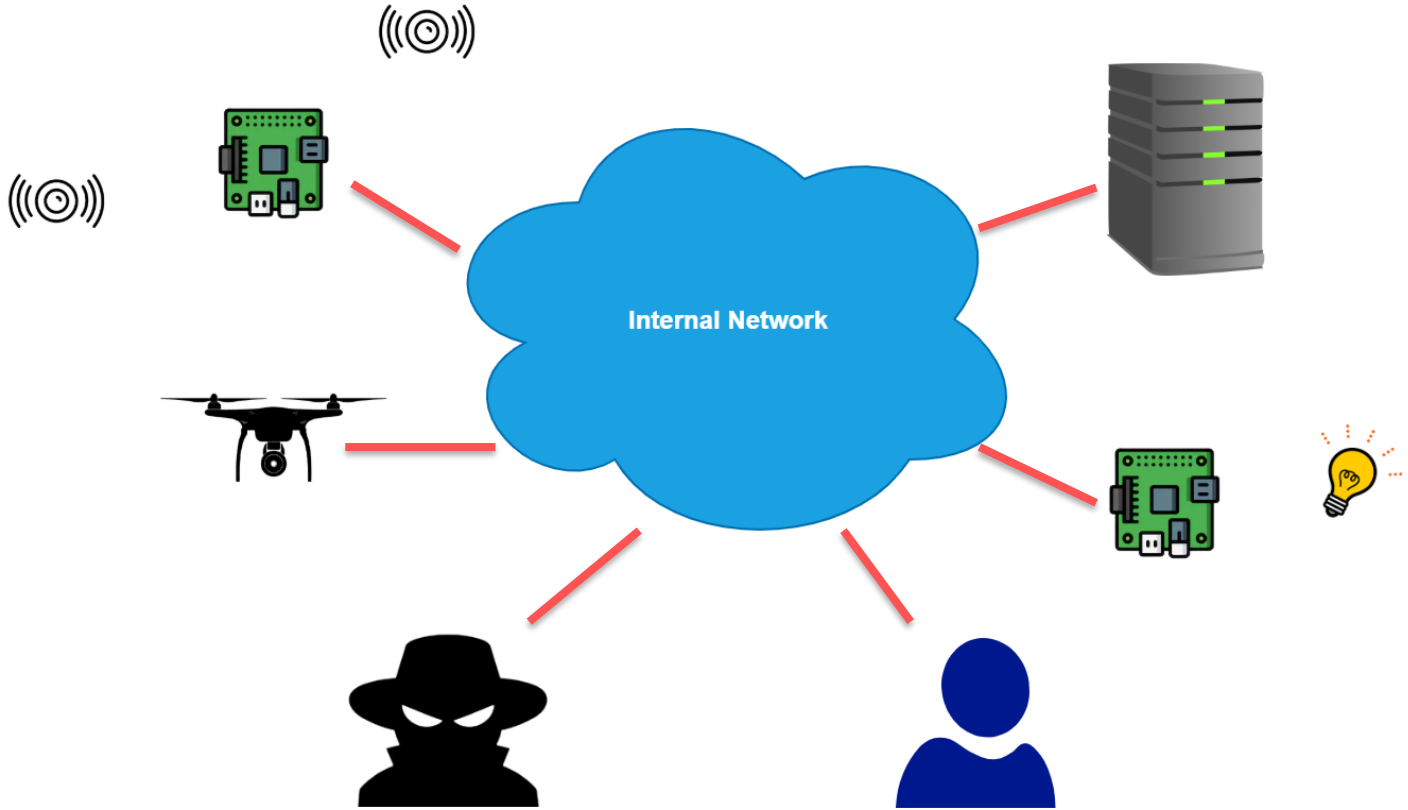
Security Issues

Confidentiality, integrity & authenticity of sensitive data stored on disk



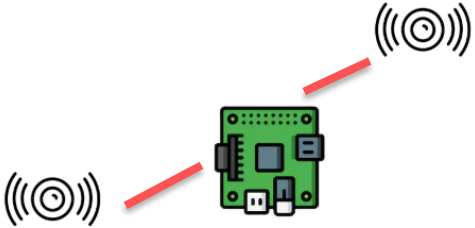
Security Issues

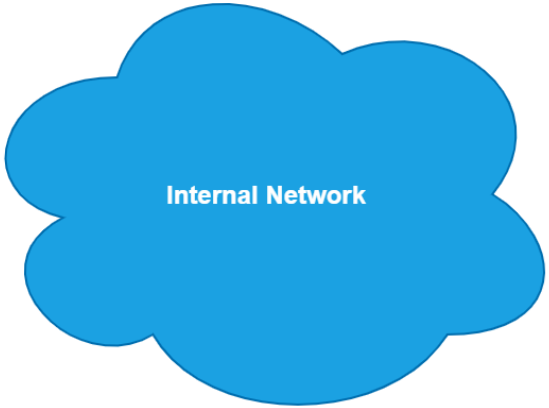
Confidentiality, integrity & authenticity of sensitive data on the network

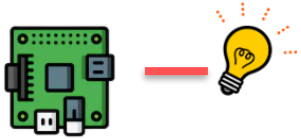


Security Issues

Integrity & authenticity of the interaction with I/O devices (bi-directional)





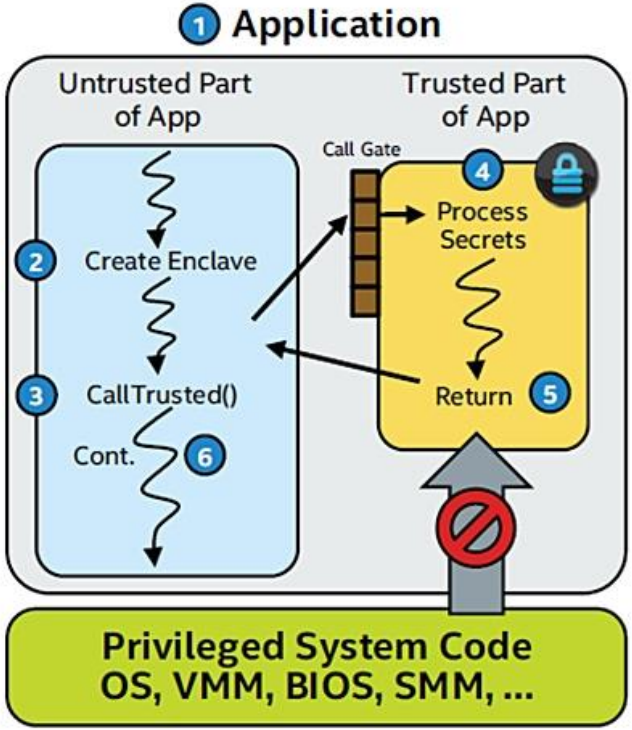


Research question

- How can we protect a system from **all** the security issues seen before?
- While, at the same time, **minimizing** the Trusted Computing Base (TCB)
- The system might be:
  - Heterogeneous
  - Distributed

Trusted Execution Environment (TEE)

- **Isolation** of sensitive code and data
- **Reduction** of the attack surface (TCB)
- **Authentication** of the running software (Remote Attestation)
- Examples of TEEs:
  - Intel Security Guard Extensions (SGX)
  - ARM TrustZone
  - Sancus



State of the art

- Trusted paths between high-end nodes and I/O devices
  - ***BitE***: I/O devices with crypto capabilities, no TEE
  - ***Bumpy***: BitE + Flicker TEE
  - ***SGXIO***, ***seL4***: SGX machines with kernel modifications
  - ***SGX-USB***: proxy devices between SGX platform and I/O peripherals
  - ***Fidelius***: SGX machines and trusted I/O devices

- These solutions **do not provide** *heterogeneity*
  - Not suitable for Smart Environments
  - No embedded TEEs used

Authentic Execution

- **Goal**: strong assurance of the correct execution of a distributed application
- Its principles can be applied to any TEE -> **heterogeneity!**
- The real implementation only supports Sancus
  - Using the Secure I/O feature

Contributions of this Thesis

- Design & Implementation of **Authentic Execution for Intel SGX**
  - Ensuring compatibility between Sancus and SGX
  - Developing tools for automatic deployment
- Design & Implementation of a **prototype** for smart irrigation
- **Evaluation** of the prototype: code, performance, security

Authentic Execution SGX

- **Design**: how to apply the high-level concepts of the paper to this context
  - Programming languages, frameworks, data structures..
  - Small extensions to the framework
  - Application-level network protocol compatible with Sancus
- **Implementation**: framework written entirely in **Rust**
  - Modern, fast programming language
  - **Secure** by design

Prototype

- Simple smart irrigation system
- Four modules:
  - Sensor
  - Actuator
  - Controller
  - Dashboard
- No real I/O devices
  - Sensor -> simulated in software
  - Actuator -> Used an LED with Secure I/O

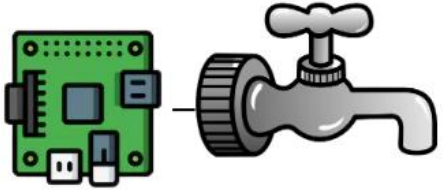
Prototype (cont'd)

Dashboard

Controller

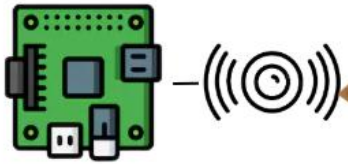
SGX

Actuator



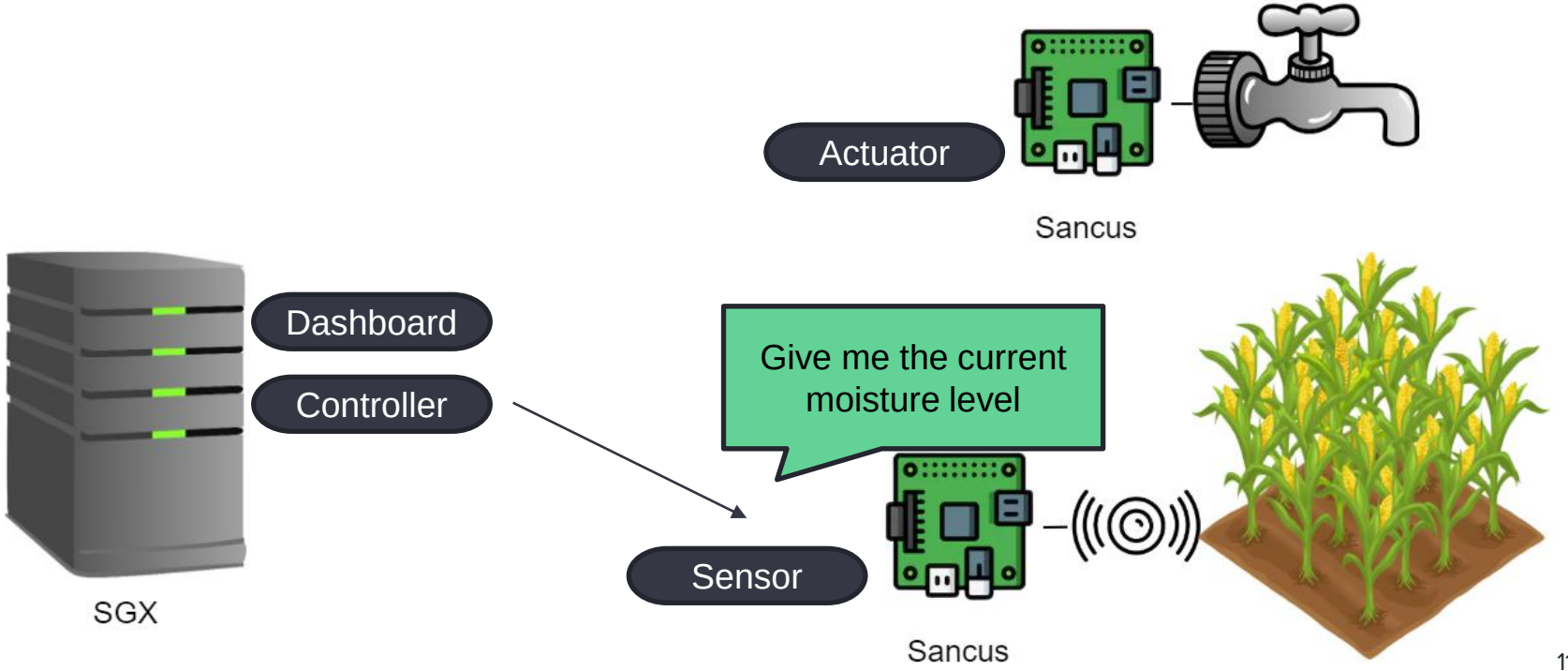
Sancus

Sensor

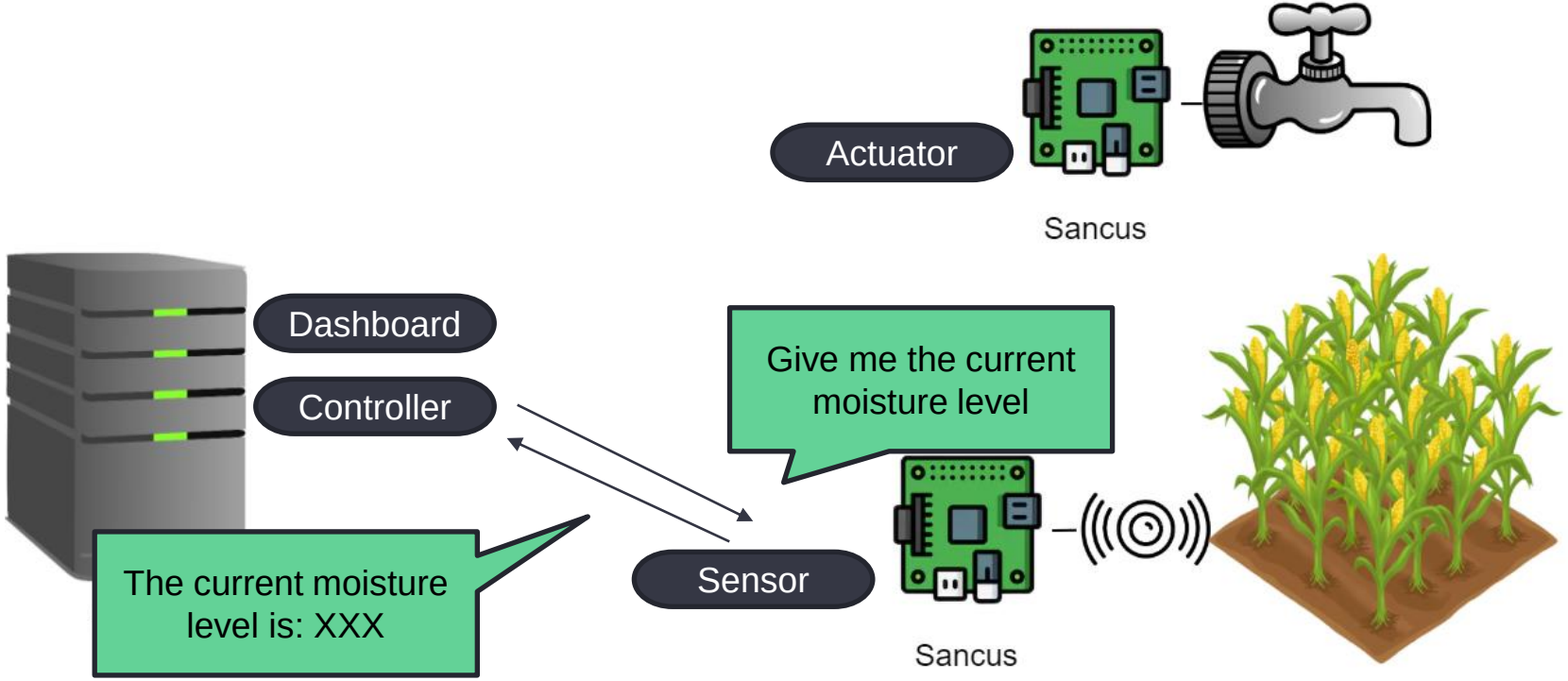


Sancus

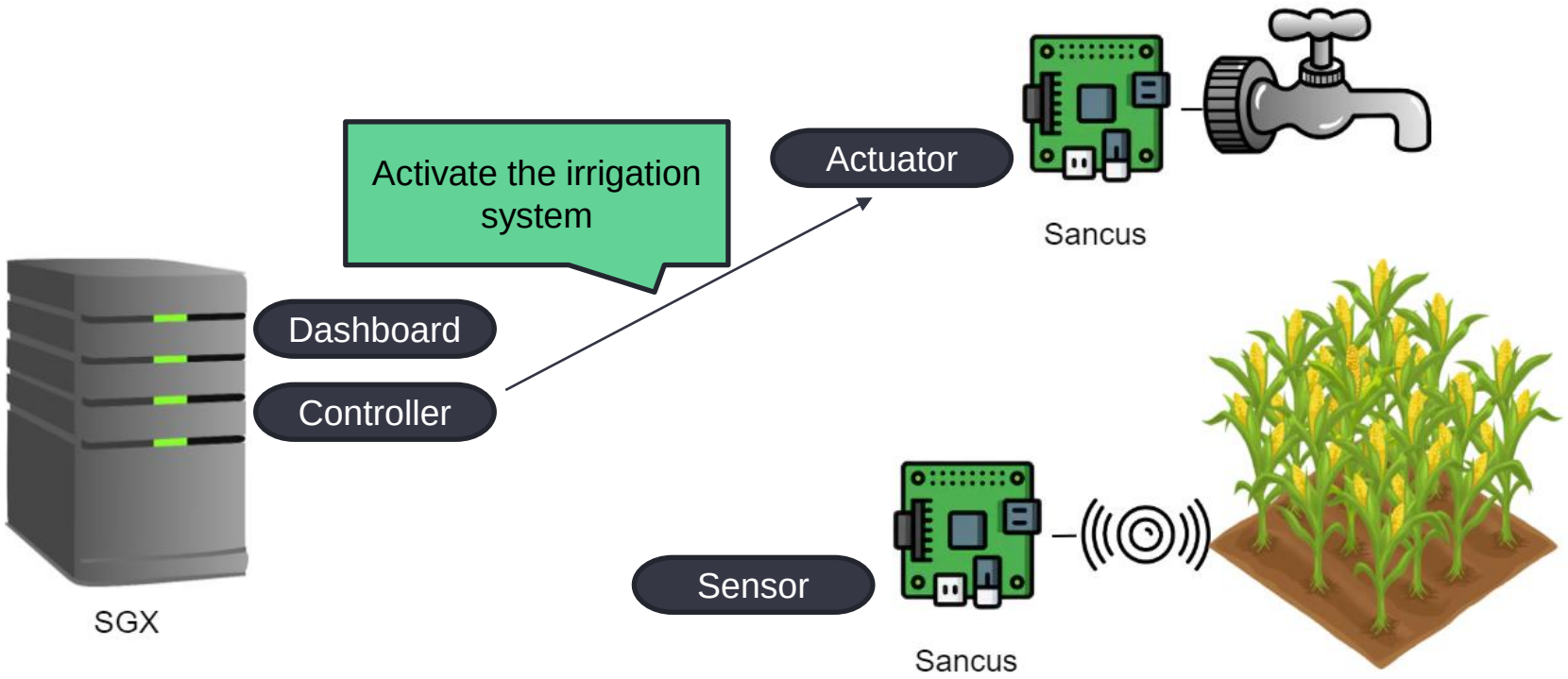
Prototype (cont'd)



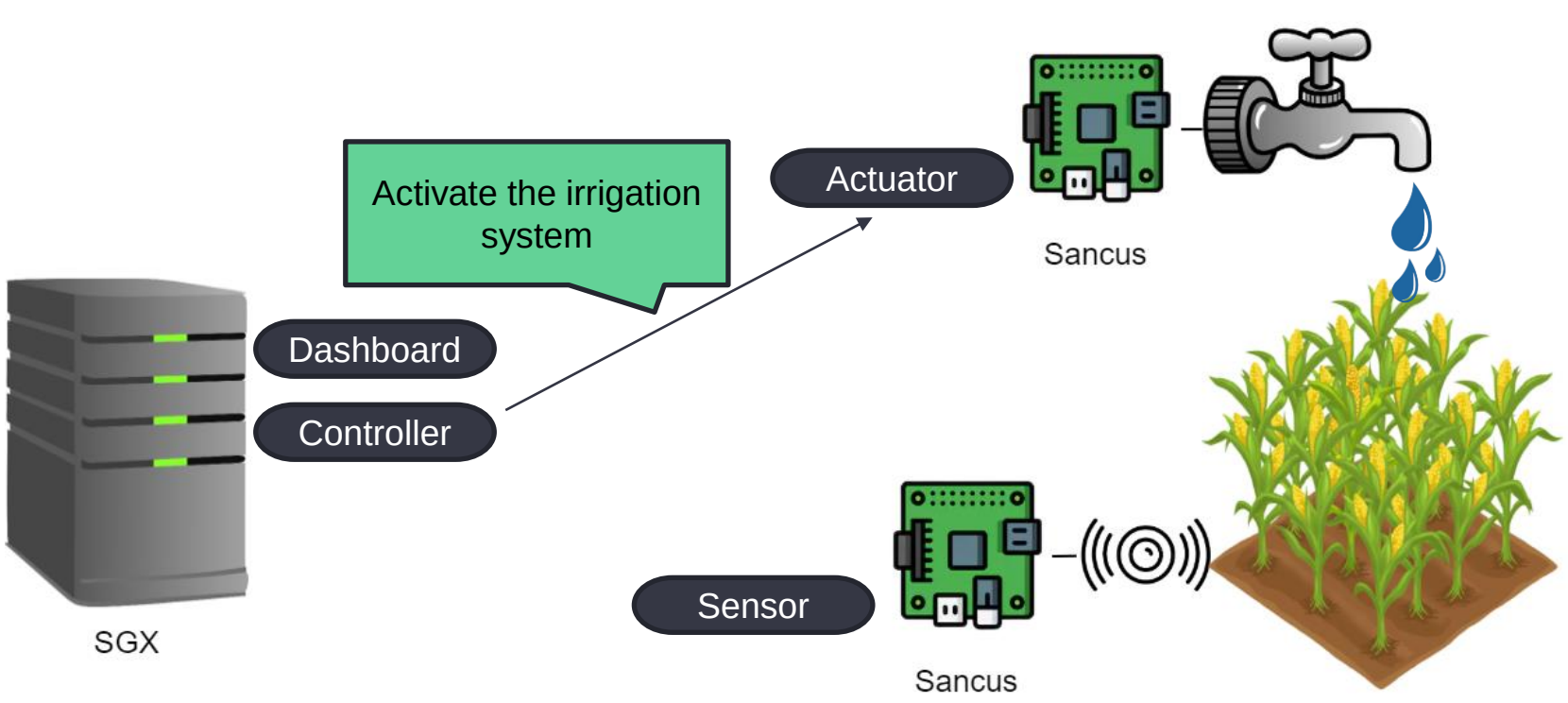
Prototype (cont'd)



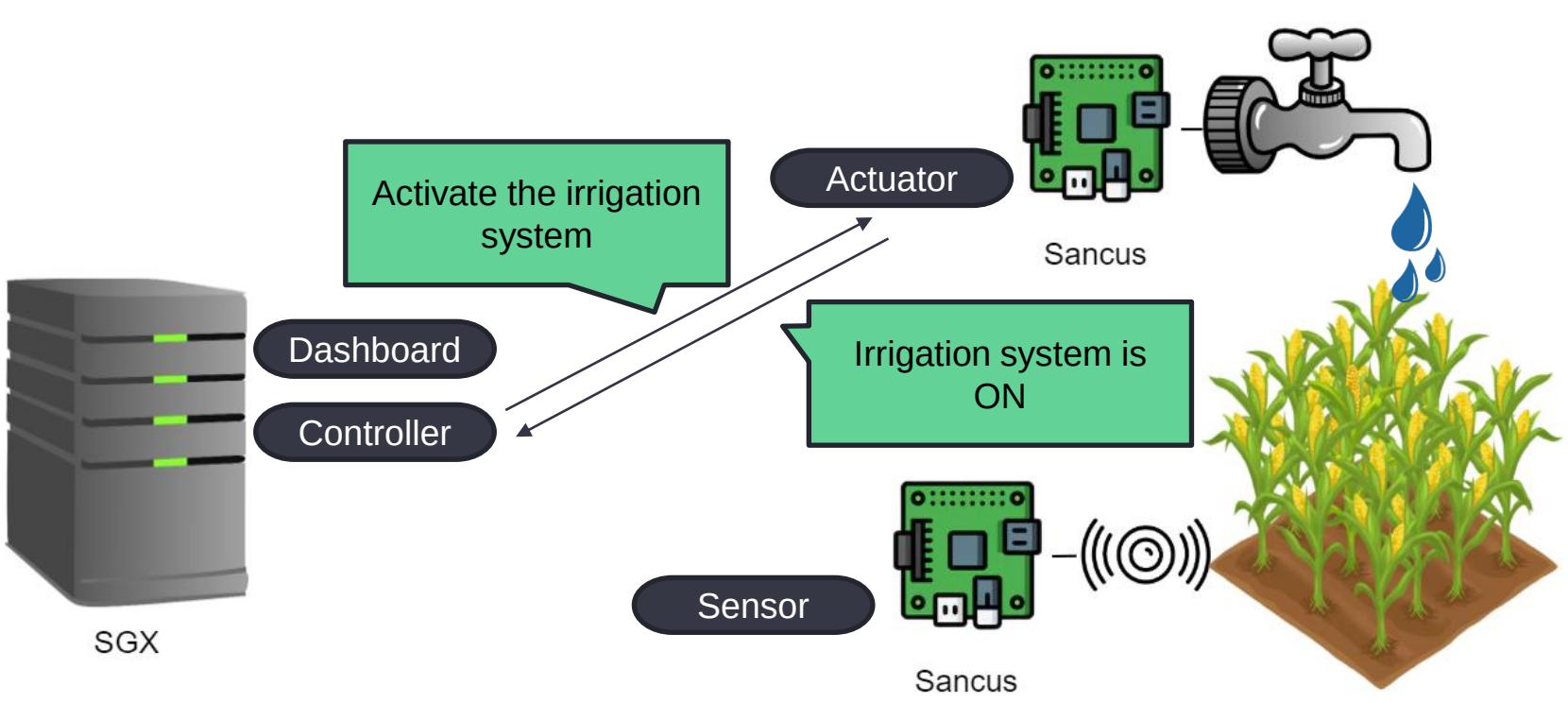
Prototype (cont'd)



Prototype (cont'd)



Prototype (cont'd)



Prototype (cont'd)

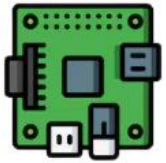
Is the irrigation system ON?

Dashboard

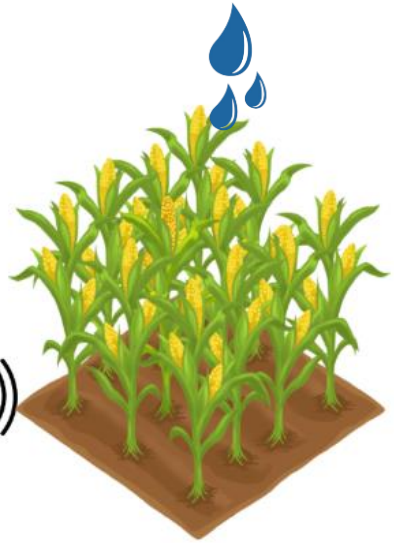
Controller

SGX

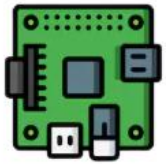
Actuator



Sancus

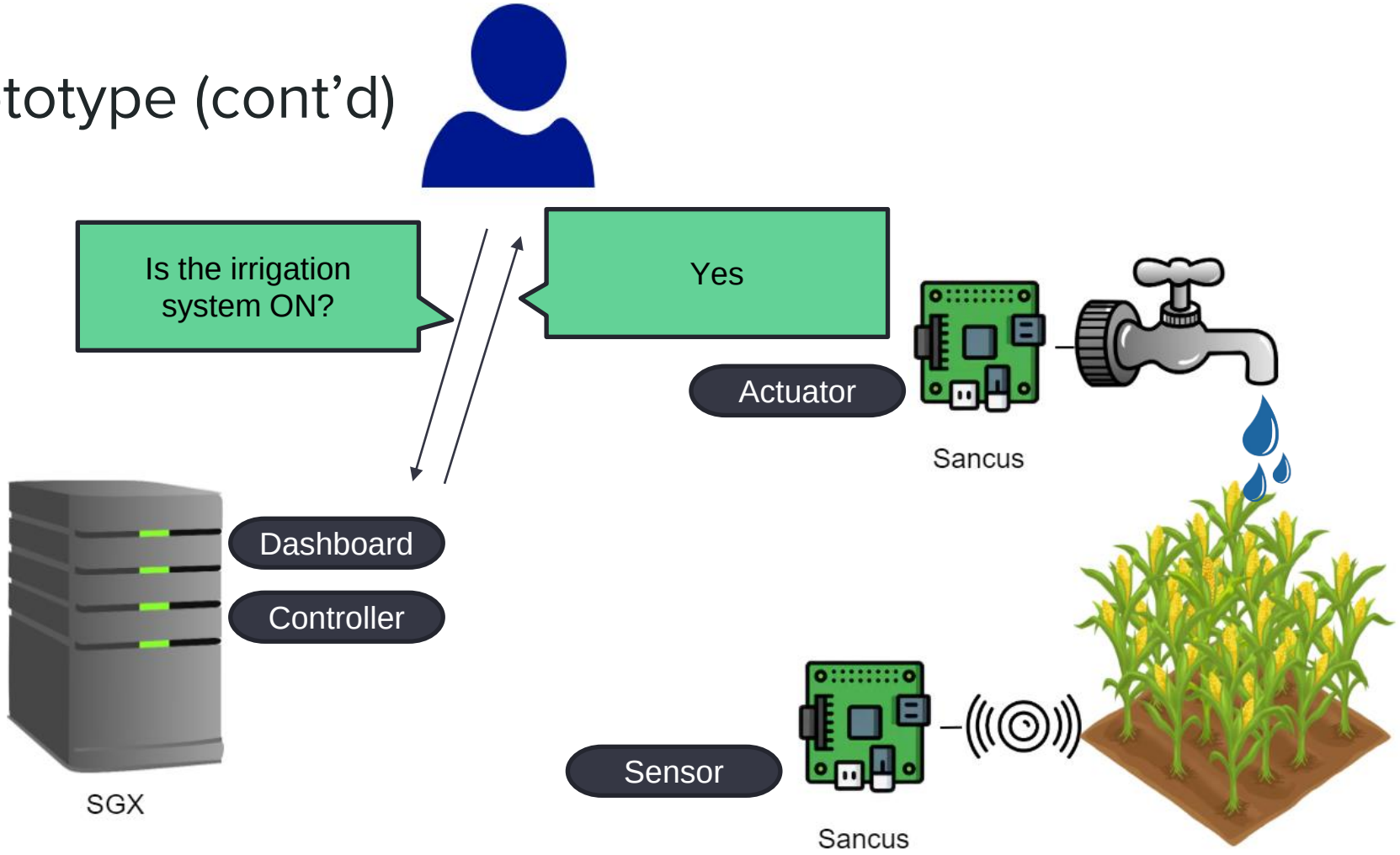


Sensor



Sancus

Prototype (cont'd)



Conclusions

- Code: very small TCB
  - Thanks to isolation mechanisms of TEEs
  - **Minimised** developer effort (between 24 and 53 SLOC for each component's logic)

- Security:
  - Strong integrity
  - Confidentiality of sensitive data

Conclusions (cont’d)

- Performance:
  - Response time in the order of ~100ms
  - **OK** smart irrigation, possible issues with stricter real time requirements

- Future work:
  - Test a complete prototype
  - Improve performance
  - Add features (e.g., SGX sealing)

Thank you for your attention!
Questions?