Authentic Execution in Smart Farming

Student: Gianluca Scopelliti

Supervisors: Frank Piessens, Jan Tobias Mühlberg, Fritz Alder

Smart Farming

What is Smart Farming?

- Application of modern ICT in agriculture
- **Embedded devices** connected to:
  - **Sensors**, to collect data from the physical world (e.g. temperature, humidity, light…)
  - **Actuators**, to perform operations to the physical world (e.g. irrigation, animal feeding…)
- **Central servers** to store data and execute software for computing statistics and/or predictions

Why?

- In general, the goal is to **maximize profits** and **minimize costs**
- The data collected is processed by servers using specialized software, providing to the farmer statistics and support for decisions
  - Typically, the farmer interacts with the system through a dashboard

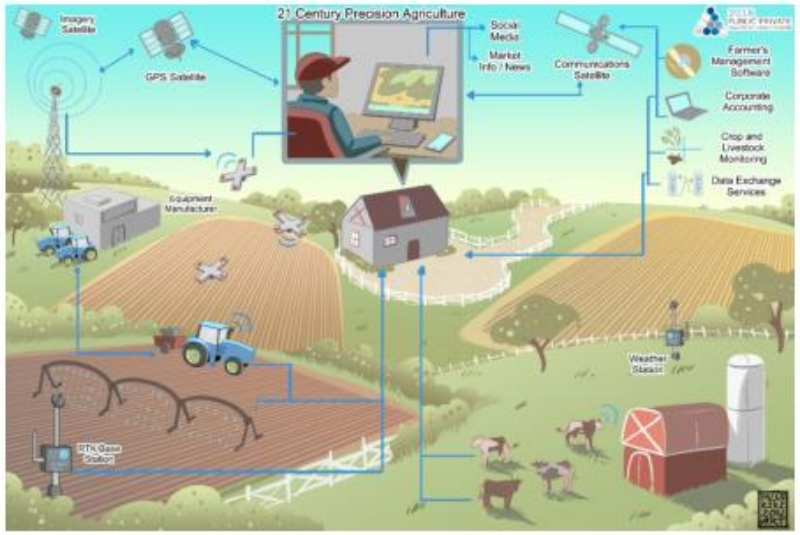
Why?

- In general, the goal is to **maximize profits** and **minimize costs**
- The data collected is processed by servers using specialized software, providing to the farmer statistics and support for decisions
  - Typically, the farmer interacts with the system through a dashboard
- The main benefits of using this approach are:
  - **Improve input efficiency**, by applying the optimal amount of nutrients, seed or chemical products such as pesticides at the right location and time, using the right type of product
  - **Identify anomalies** such as diseases, by analyzing the general health of the crop and livestock
  - **Reduce labor time and costs,** by using actuators and specialized software for performing operations with limited human intervention

Threats to Precision Agriculture

2018 Public-Private Analytic Exchange Program

Threats to Precision Agriculture



- Paper released by US Department of Homeland Security (DHS) in 2018

- *"[…] addresses the security threats related to the adoption and impact of new digital technologies in crop and livestock production."*

Security issues

Security issues

- **Confidentiality**
  - Theft of data
  - Leak of data
  - Foreign access to data

Security issues

- **Confidentiality**
  - Theft of data
  - Leak of data
  - Foreign access to data
- **Integrity**
  - Manipulation of data
  - Introduction of fake data
  - Manipulation of actuators

Security issues

- **Confidentiality**
  - Theft of data
  - Leak of data
  - Foreign access to data
- **Integrity**
  - Manipulation of data
  - Introduction of fake data
  - Manipulation of actuators
- **Availability**
  - Block / alter nodes
  - Block / alter communication between nodes

Trusted Computing and Authentic Execution

Trusted Computing

- Technology developed by the ***Trusted Computing Group***
- Goal: provide strong guarantees that a software will **not** misbehave
  - Even in a system where all the other components (OS, other SW) are untrusted!

Trusted Computing

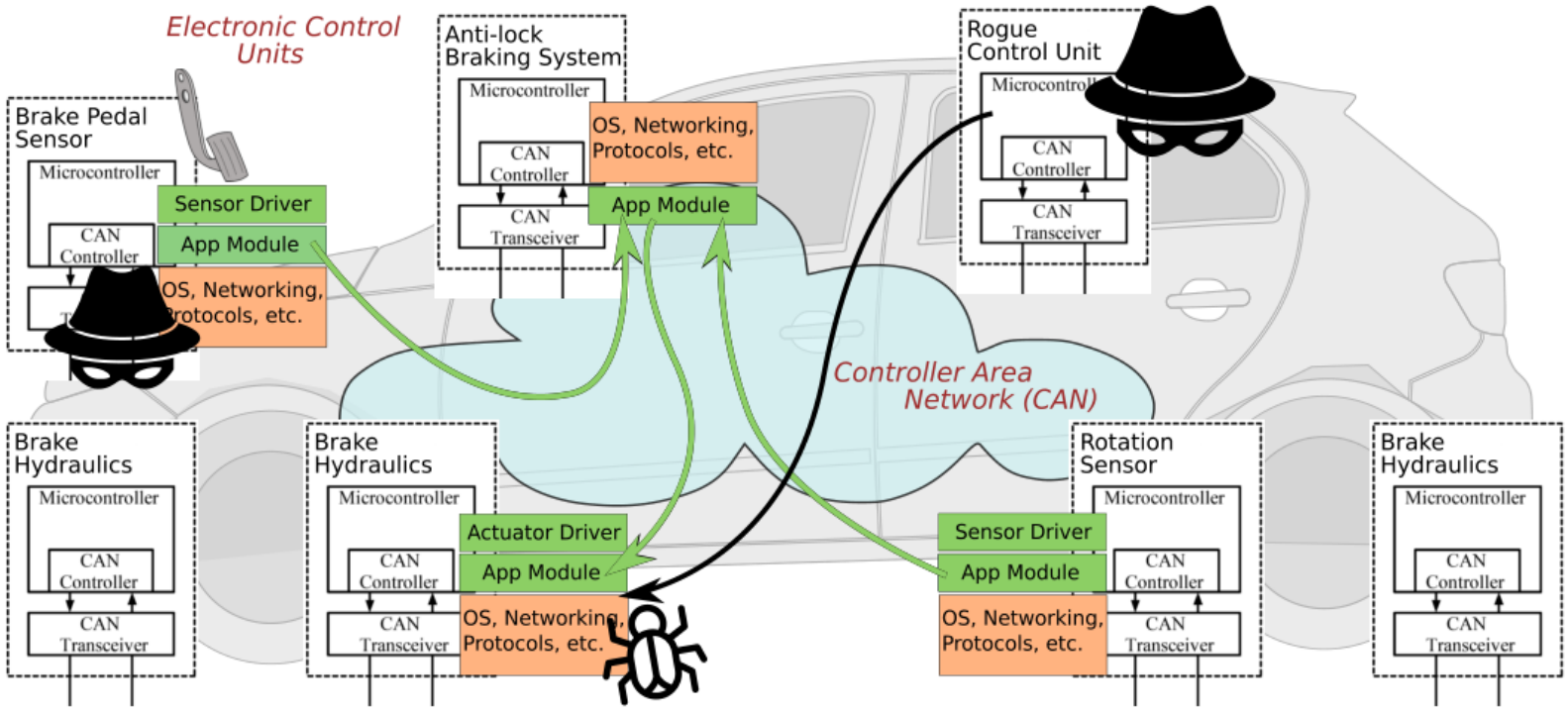
- Technology developed by the ***Trusted Computing Group***
- Goal: provide strong guarantees that a software will **not** misbehave
  - Even in a system where all the other components (OS, other SW) are untrusted!
- Key concepts:
  - **Endorsment Key**: unique private key that never leaves the hardware
  - **Memory curtaining**: isolation of sensitive areas of memory
  - **Sealed storage**: bind data to a specific device or software
  - **Remote attestation**: authenticate hw / sw configuration to a remote host
  - **Trusted third party**: an intermediary to provide (ano|pseudo)nymity

Trusted Computing

- Technology developed by the ***Trusted Computing Group***
- Goal: provide strong guarantees that a software will **not** misbehave
  - Even in a system where all the other components (OS, other SW) are untrusted!
- Key concepts:
  - **Endorsment Key**: unique private key that never leaves the hardware
  - **Memory curtaining**: isolation of sensitive areas of memory
  - **Sealed storage**: bind data to a specific device or software
  - **Remote attestation**: authenticate hw / sw configuration to a remote host
  - **Trusted third party**: an intermediary to provide (ano|pseudo)nymity
- In practice we have many different architectures which implement a subset of these features, and they also provide new features such as *Enclaved execution* and *secure I/O*

VulCAN

- An application of Trusted Computing in the automotive context
- Force points:
  - Message authentication
  - Software component attestation and isolation



Authentic Execution

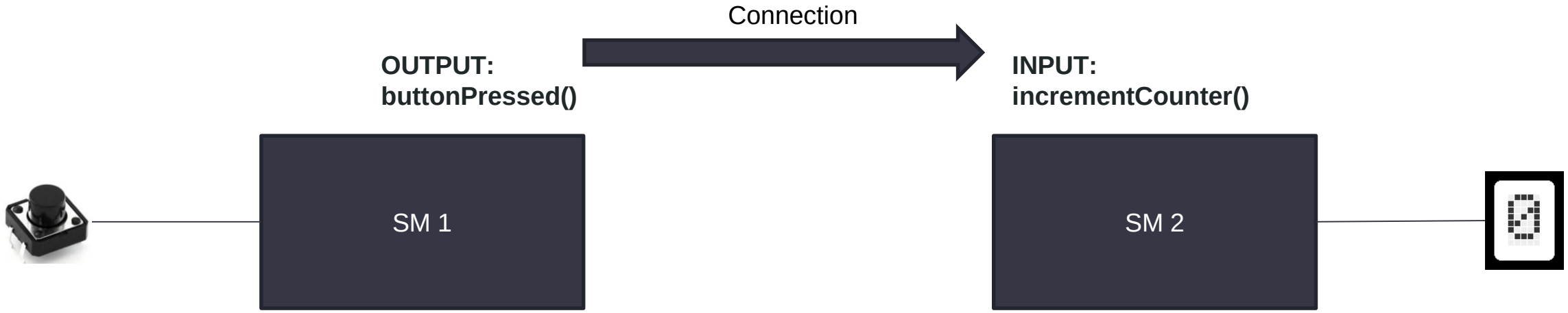
«if the application produces a physical output event (e.g., turns on an LED), then there must have happened a sequence of physical input events such that that sequence, when processed by the application (as specified in the high-level source code), produces that output event. »

Authentic Execution

«if the application produces a physical output event (e.g., turns on an LED), then there must have happened a sequence of physical input events such that that sequence, when processed by the application (as specified in the high-level source code), produces that output event. »

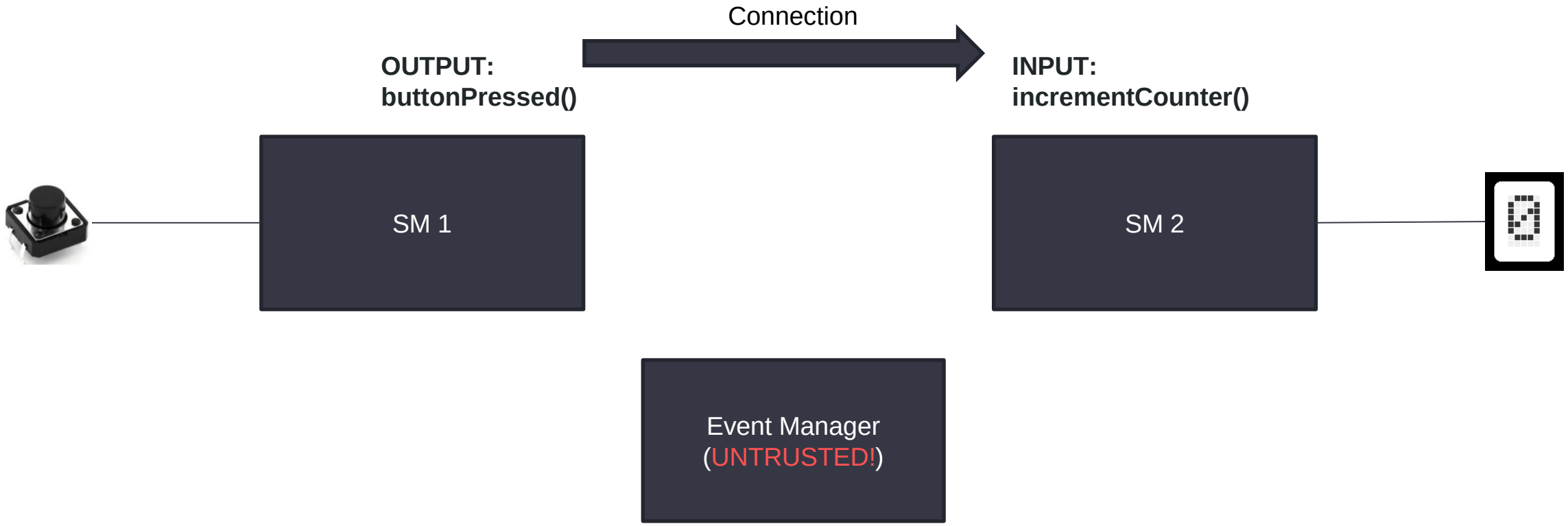
Authentic Execution

«if the application produces a physical output event (e.g., turns on an LED), then there must have happened a sequence of physical input events such that that sequence, when processed by the application (as specified in the high-level source code), produces that output event. »



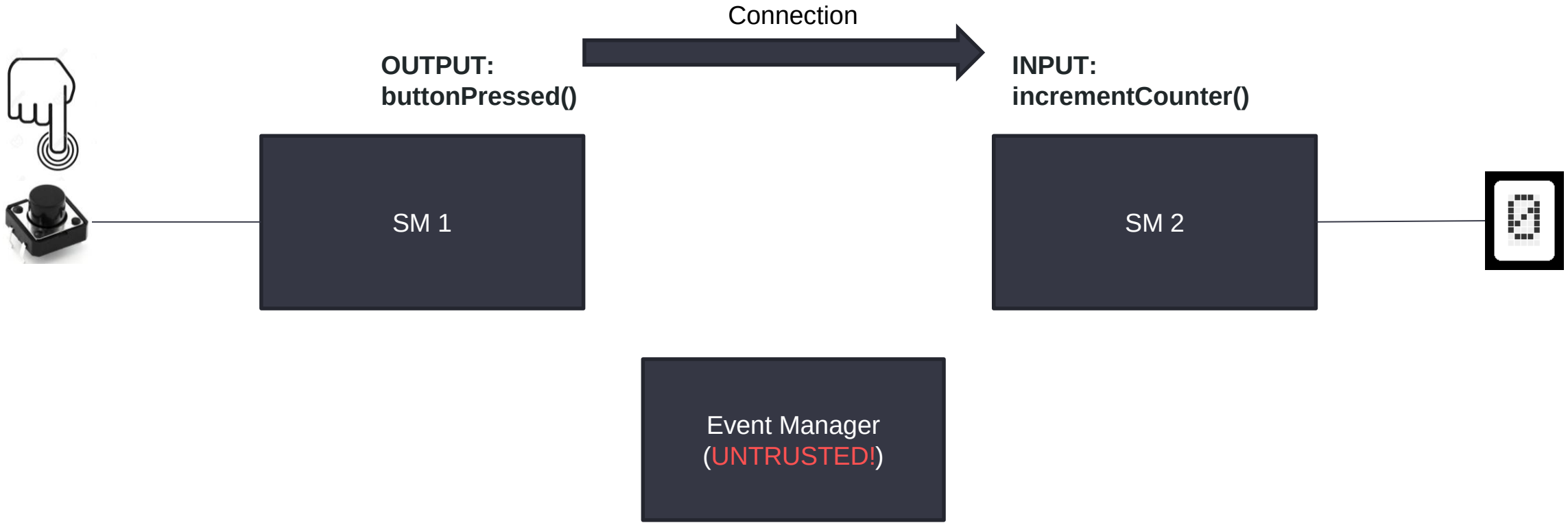
Authentic Execution

«if the application produces a physical output event (e.g., turns on an LED), then there must have happened a sequence of physical input events such that that sequence, when processed by the application (as specified in the high-level source code), produces that output event. »



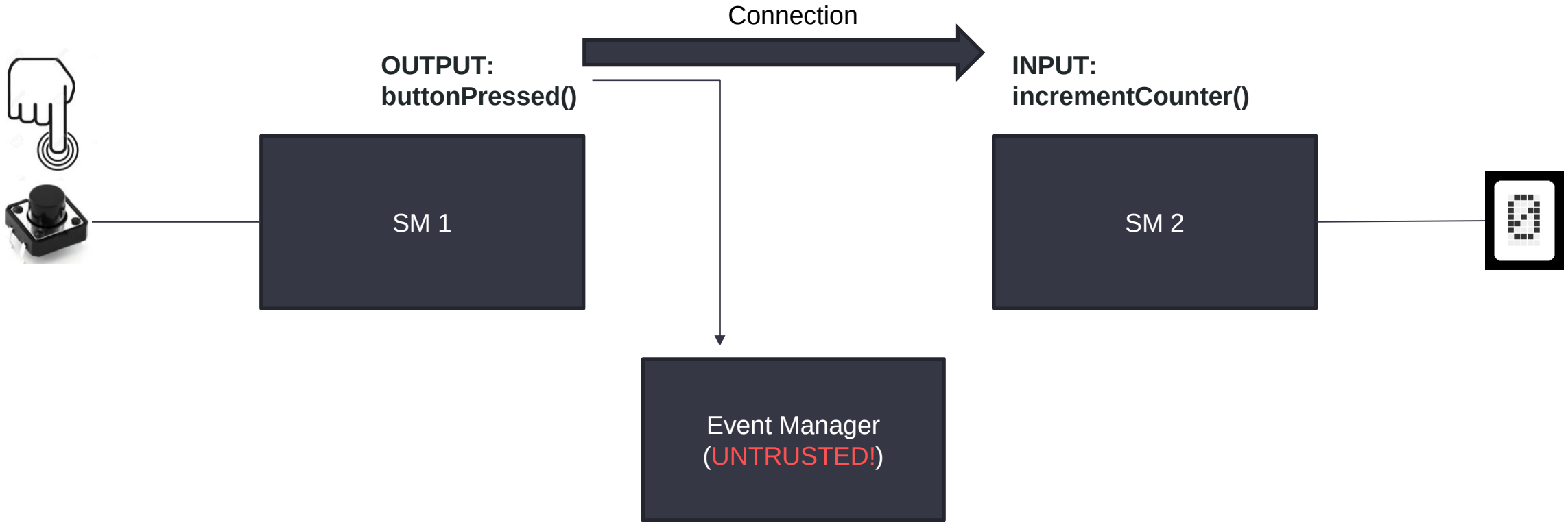
Authentic Execution

«if the application produces a physical output event (e.g., turns on an LED), then there must have happened a sequence of physical input events such that that sequence, when processed by the application (as specified in the high-level source code), produces that output event. »



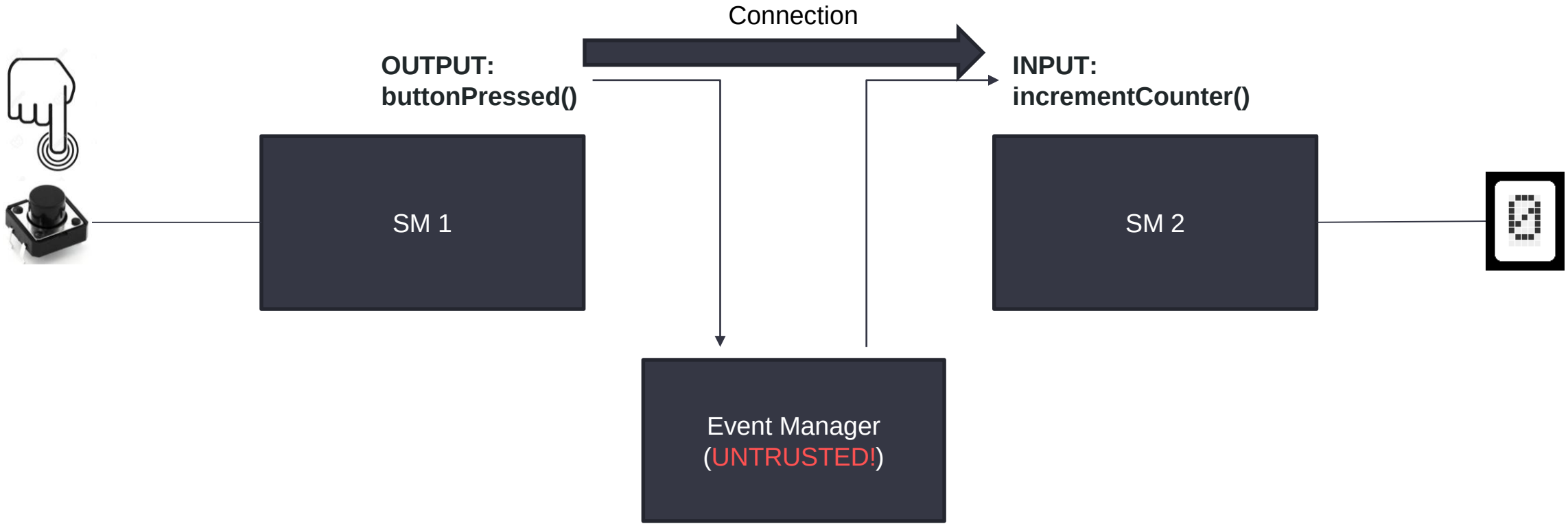
Authentic Execution

«if the application produces a physical output event (e.g., turns on an LED), then there must have happened a sequence of physical input events such that that sequence, when processed by the application (as specified in the high-level source code), produces that output event. »



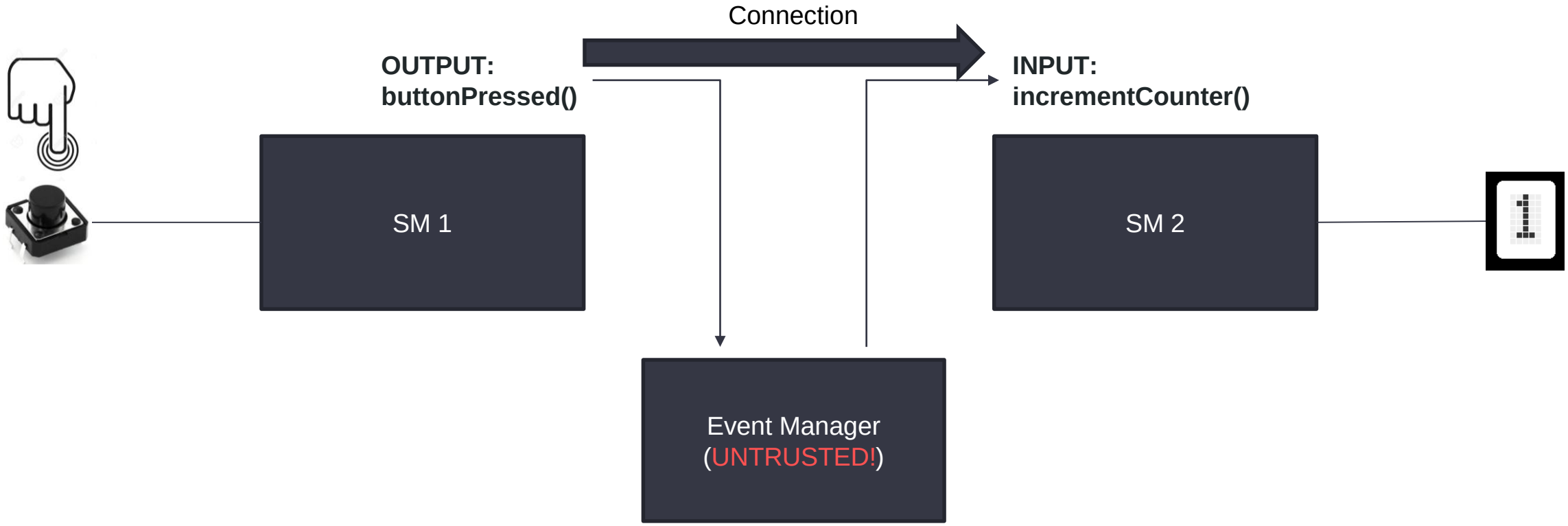
Authentic Execution

«if the application produces a physical output event (e.g., turns on an LED), then there must have happened a sequence of physical input events such that that sequence, when processed by the application (as specified in the high-level source code), produces that output event. »



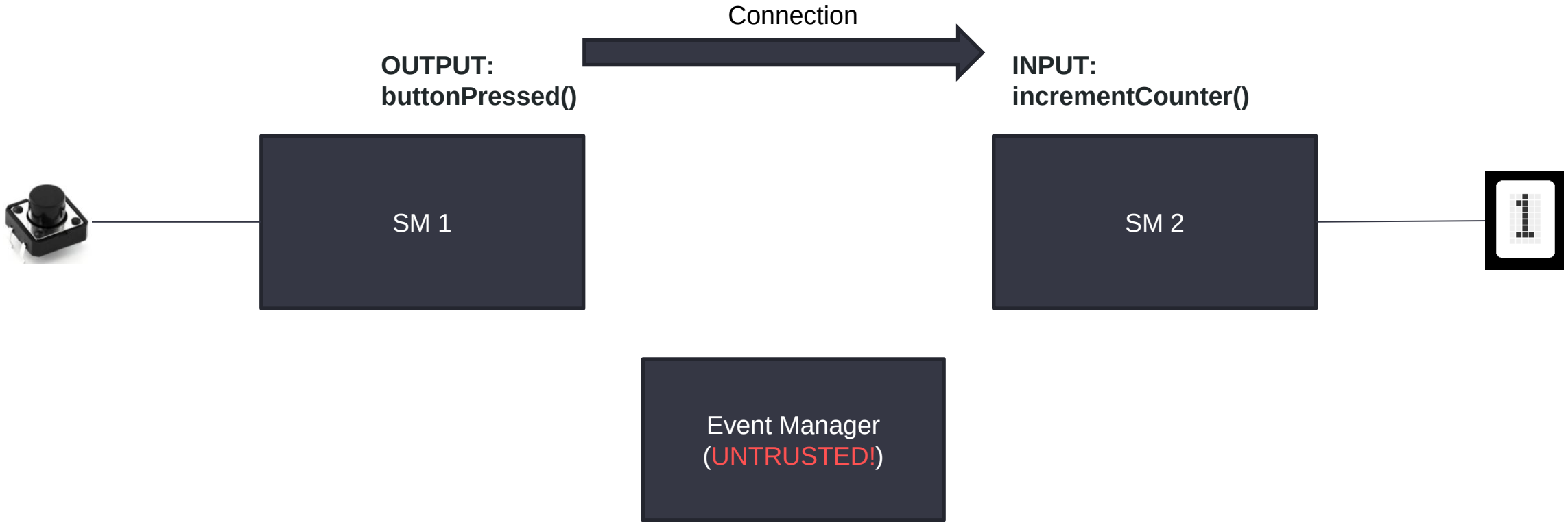
Authentic Execution

«if the application produces a physical output event (e.g., turns on an LED), then there must have happened a sequence of physical input events such that that sequence, when processed by the application (as specified in the high-level source code), produces that output event. »



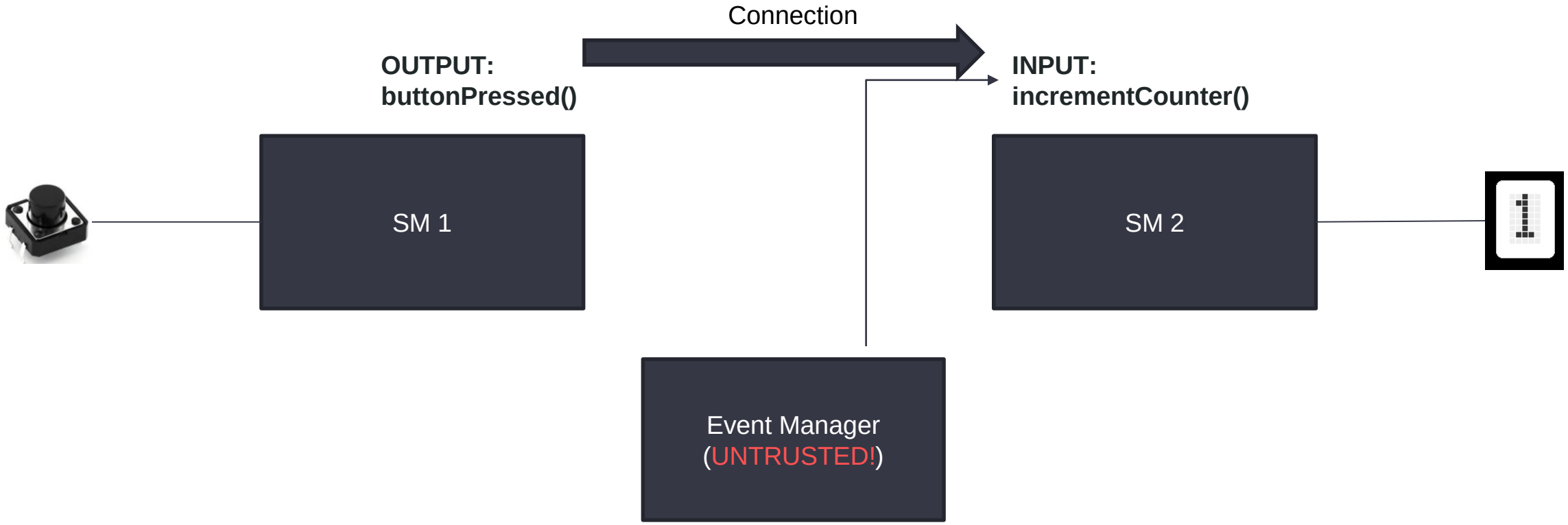
Authentic Execution

«if the application produces a physical output event (e.g., turns on an LED), then there must have happened a sequence of physical input events such that that sequence, when processed by the application (as specified in the high-level source code), produces that output event. »



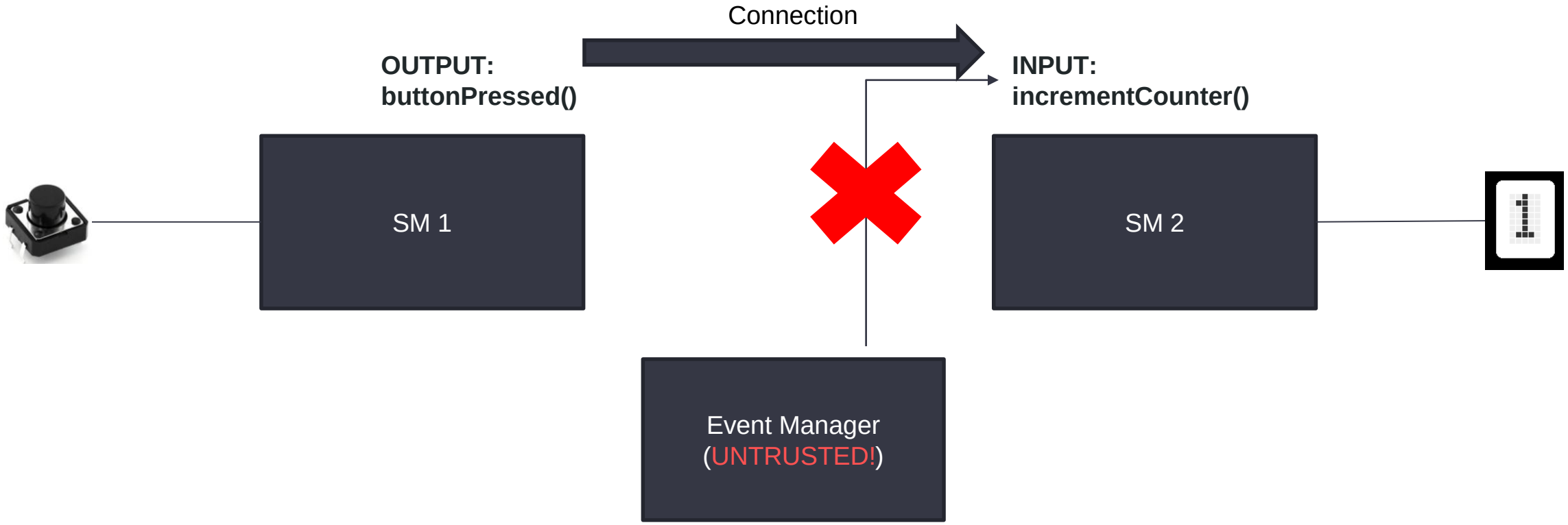
Authentic Execution

«if the application produces a physical output event (e.g., turns on an LED), then there must have happened a sequence of physical input events such that that sequence, when processed by the application (as specified in the high-level source code), produces that output event. »



Authentic Execution

«if the application produces a physical output event (e.g., turns on an LED), then there must have happened a sequence of physical input events such that that sequence, when processed by the application (as specified in the high-level source code), produces that output event. »



Limitations of the current implementations

- VulCAN handles a **specific** communication medium (CAN)
  - The code is then adapted for dealing with its characteristics
    - Broadcast network
    - 64-bit packets (-> MAC truncation)
    - …
- The current implementation of Authentic Execution manages only Sancus devices
  - What about different architectures like SGX?

My Master’s Thesis

My Master’s Thesis work

- Generalize the concept of Authentic Execution

My Master's Thesis work

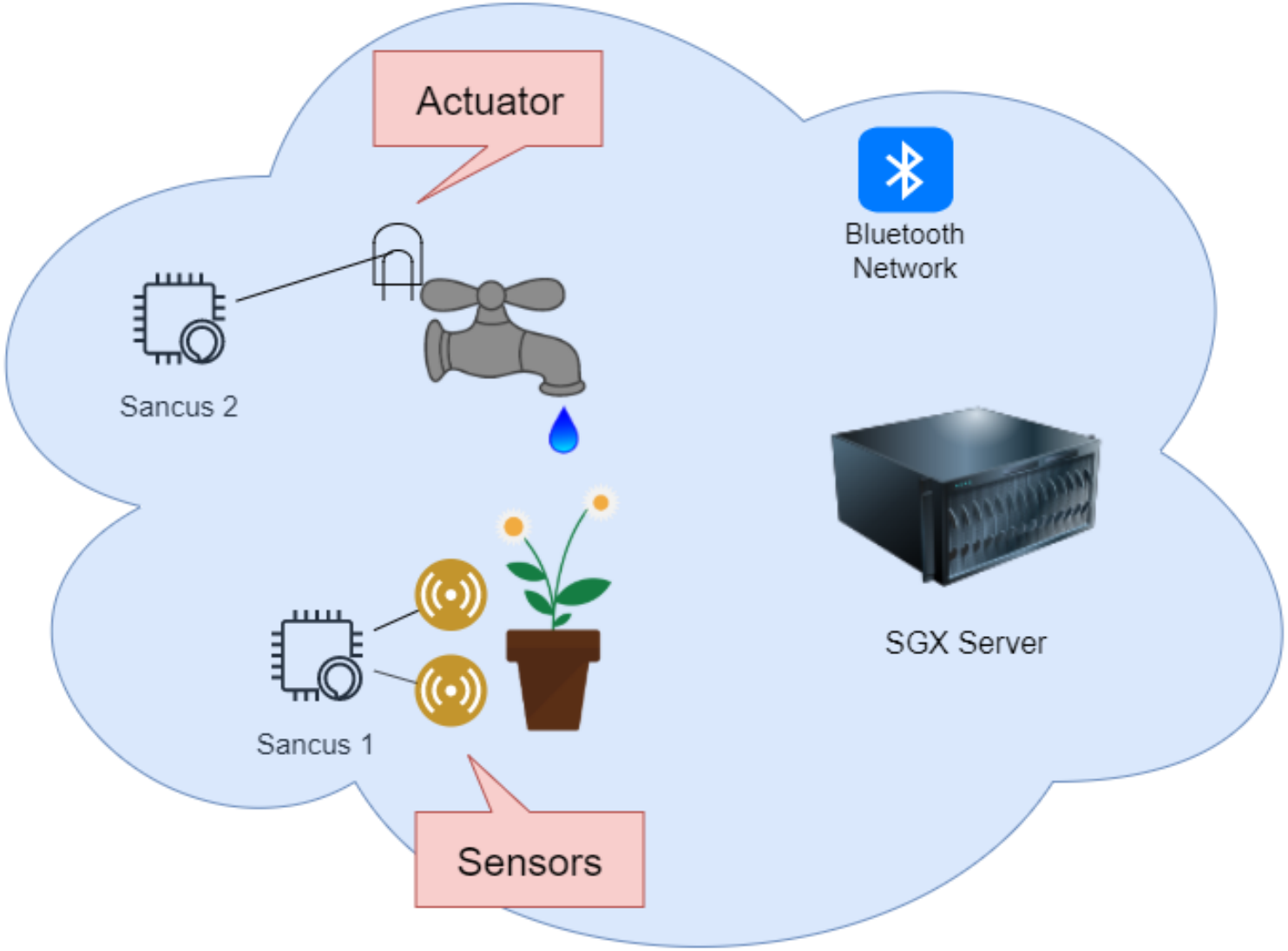
- Generalize the concept of Authentic Execution
  - Communication medium: provide a sort of ***generic API*** which abstracts the network layer
    - Goal: the code written by the developer should be **transparent** to the communication medium used
    - The developer has only to specify the technology used and its parameters in a configuration file

My Master's Thesis work

- Generalize the concept of Authentic Execution
  - Communication medium: provide a sort of ***generic API*** which abstracts the network layer
    - Goal: the code written by the developer should be **transparent** to the communication medium used
    - The developer has only to specify the technology used and its parameters in a configuration file
  - Different architectures: provide support for another architecture (Intel SGX)
    - Goal: we want to simulate a system composed by heterogeneous components (from lightweight IoT devices to desktop/server architectures)
      - Many real scenarios present such situation (e.g. Smart Farming)

A prototype: Authentic Execution for automatic irrigation

- **Goal**: apply the concepts described in the previous slide to a concrete use case in the context of Smart Farming



Schedule

- Within the holidays: completing the reading / planning phase
- From January onwards: implementation phase
  - Authentic execution between two SGX enclaves
  - Authentic execution in a heterogeneous system (SGX + Sancus)
  - Generic API for the communication

Conclusions

Conclusions

- Goal: Create a system which provides strong security properties in presence of malicious actors
  - Confidentiality, integrity features
- Application in a specific context (smart farming)..
  - .. but can be also applied to any other areas with different:
    - Architectures
    - Communication mediums

References

- *Threats to Precision Agriculture*
- Tutorial dsn18 slides
- Article about Smart Farming
- VulCAN
- Authentic Execution
- Image slide 3
- Image slide 6