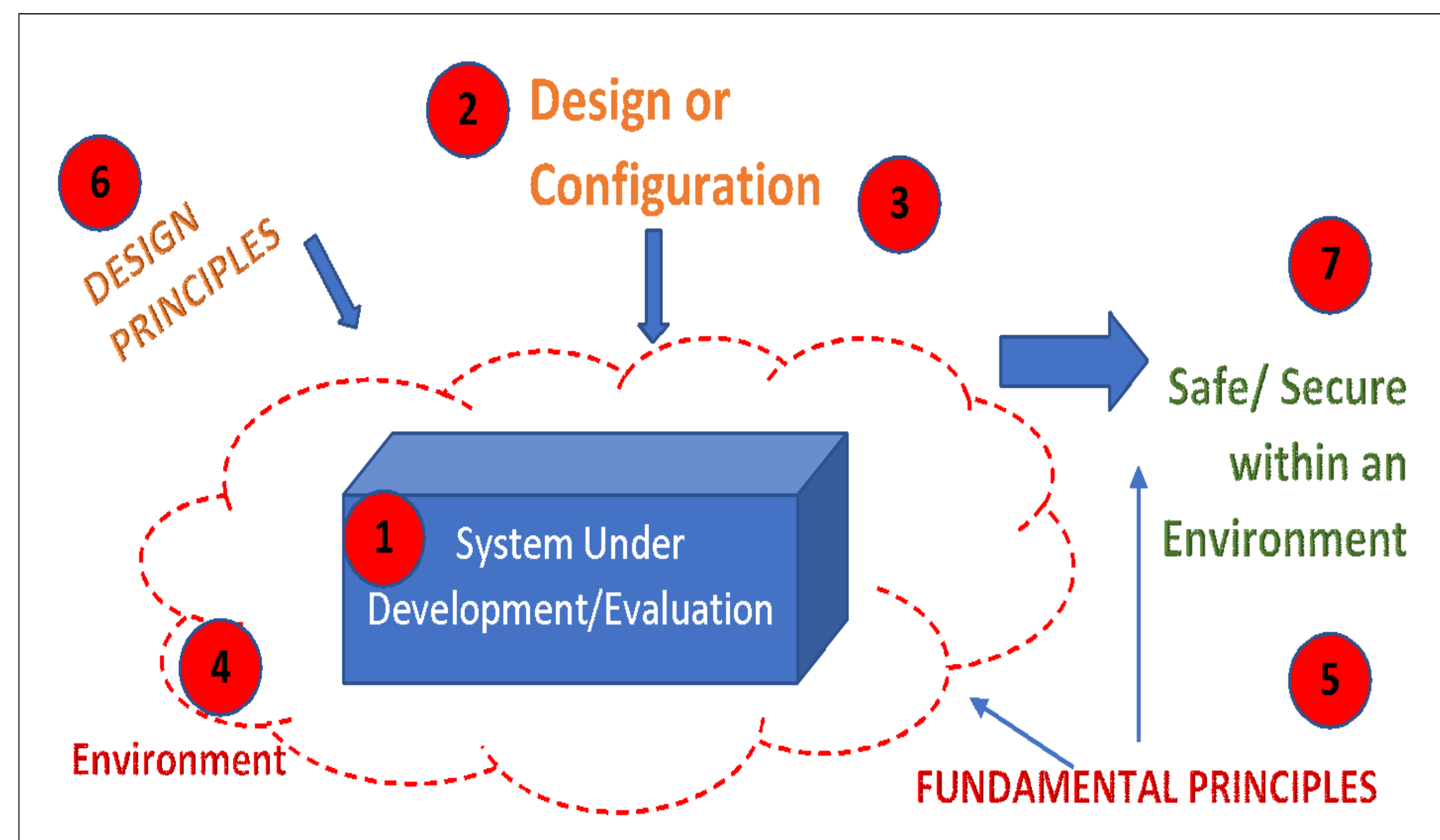


# Building Secure Embedded Control Systems



The central system (1) in this diagram represents the system under development/evaluation that we wish to consider.

- The system is designed (2) to meet certain operational characteristics.
- This system can be further configured (3) to limit operation. This is the normal state of operation for most systems.
- The system, when deployed in a particular environment (4) operates following the laws of physics and other fundamental principles (5) of operation.
- Experience, sound engineering practices, and science can provide us with design principles (6) to create a system that meets our requirements.
- Using our understanding of the fundamental principles and based on experience and science we can analyze the system (7) and understand the context in which it is safe/secure or meets other criteria.

## Project Overview

The goal of this project is to explore the concepts of cybersecurity such that we can understand under which design constraints can we build a system that is intrinsically secure. We are exploring theoretical, software and hardware aspects of secure computing to develop that answer in the context of an embedded control system.

### Keywords of Progress:

Memory Tagging, Pointer Security, Weakest Precondition Calculation, Language-Based Solutions

## What do we mean by secure?

### Security as Written

This involves analysis of the program to guarantee that it will execute as intended and that outside user inputs can not dynamically change the program's behavior. This counters the classic memory corruption vulnerabilities.

### Security Between

This involves the analysis of a program's/process's interaction with other programs/processes. This includes direct changes to the memory or state of another process and controlling the passing of messages/parameters.

### Security Below

This is a guarantee that the libraries, system services and operating system behave in a manner as expected so that they can not impact the expected behavior or security of the running program. And the running program can not misuse them. The property allows us to build a system relying on other components while maintaining our overall security.

### Security as an Application

This involves analysis of the security policies and properties of the application in the context domain it is meant to support.

## Key Subprojects

### Fundamentals of Security

We are examining formal models of security fundamentals. Are there *first principles* of security. A first start: What do we mean by secure (as seen in the middle box).

### Memory Tagging / Typed Assembly

Assembly level code exists without many of the typing and other organization constructs of higher-level languages. As such, memory safety violations can occur as data values are misused. We are developing a typed assembly language to provide more inherent security features at the hardware level.

### Pointer Security

A key aspect of typed tagging is pointer tags, specifically pointers to collections of objects (arrays, struct, classes, buffers), that are traditionally prone to attack. We are exploring efficient, hardware and software based mechanisms for these tags.

### Weakest Precondition Calculation

We're exploring formal approaches to validate our solutions. Triple axiomatic semantics' weakest preconditions are used to prove program correctness. We are implementing the weakest precondition at the assembly language level, which requires some innovations over traditional approaches. The first target of our evaluation will be an analysis of system calls being developed in a companion project.

### Language-Based Solutions


We're evaluating programming language mechanisms to describe and enforce information-flow policies. Research in language-based security could focus on expressiveness, concurrency, covert channels, and security policies. Low-level security-typed target languages need to be more expressive to be supported. We're looking at building on lessons learned on related research to enable developers to specify higher layered security policies easily, in the context of ICS, where we can bound our domain.

## Research Team

Fatema Islam Meem, Aditi Pokharel, Ronisha Shigdel, Mina Soltani Siapoush, Sabiha Jannath Tisha, Jim Alves-Foss, Jia Song, and Daniel Conte de Leon

 [www.uidaho.edu/csds](http://www.uidaho.edu/csds)

 [csds@uidaho.edu](mailto:csds@uidaho.edu)

 (208) 885-4114



University of Idaho