

mStack

Introduction

Content

- Why Event-Driven
- What is mStack
- Event-Driven Core
- State Machine Core
- Console
- Conclusion

Embedded Programming

- **Use mainly sequential programming**
- Lack of good code structure
- Spaghetti code everywhere
- Limited resources: CPU, RAM, STORAGE, POWER...
- Hard to debug and maintain

Embedded Programming

- Use mainly sequential programming
- **Lack of good code structure**
- Spaghetti code everywhere
- Limited resources: CPU, RAM, STORAGE, POWER...
- Hard to debug and maintain

Embedded Programming

- Use mainly sequential programming
- Lack of good code structure
- **Spaghetti code everywhere**
- Limited resources: CPU, RAM, STORAGE, POWER...
- Hard to debug and maintain

Embedded Programming

- Use mainly sequential programming
- Lack of good code structure
- Spaghetti code everywhere
- **Limited resources: CPU, RAM, STORAGE, POWER...**
- Hard to debug and maintain

Embedded Programming

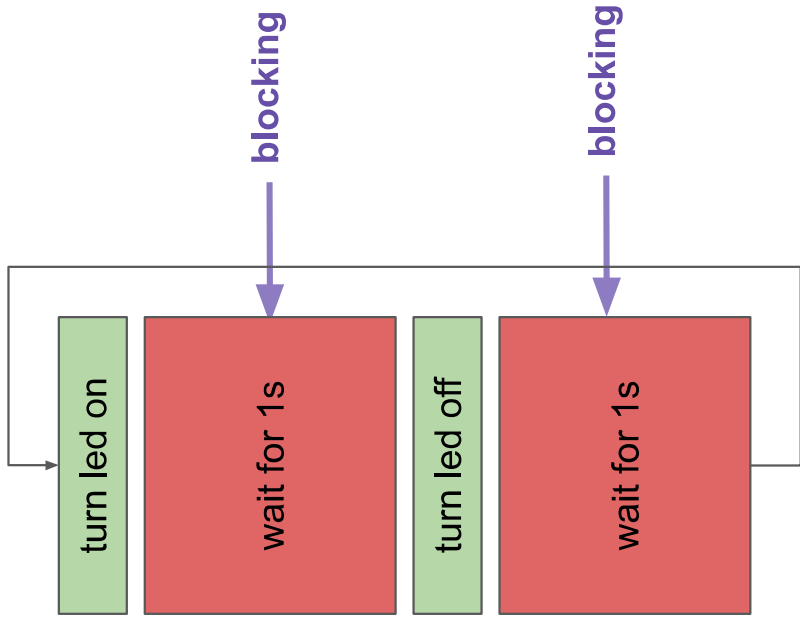
- Use mainly sequential programming
- Lack of good code structure
- Spaghetti code everywhere
- Limited resources: CPU, RAM, STORAGE, POWER...
- **Hard to debug and maintain**

Sequential Programming with Superloop

```
void main()
{
    pinMode(led, OUTPUT);
    while (1)
    {
        digitalWrite(led, HIGH);
        delay(1000);
        digitalWrite(led, LOW);
        delay(1000);
    }
}
```


Sequential Programming with Superloop

```
void main()
{
    pinMode(led, OUTPUT);
    while (1)
    {
        digitalWrite(led, HIGH);
        delay(1000);
        digitalWrite(led, LOW);
        delay(1000);
    }
}
```



Sequential Programming with RTOS

```
void thread_blink()  
{  
    pinMode(led, OUTPUT);  
    while (1)  
    {  
        digitalWrite(led, HIGH);  
        RTOS_delay(1000);  
        digitalWrite(led, LOW);  
        RTOS_delay(1000);  
    }  
}
```

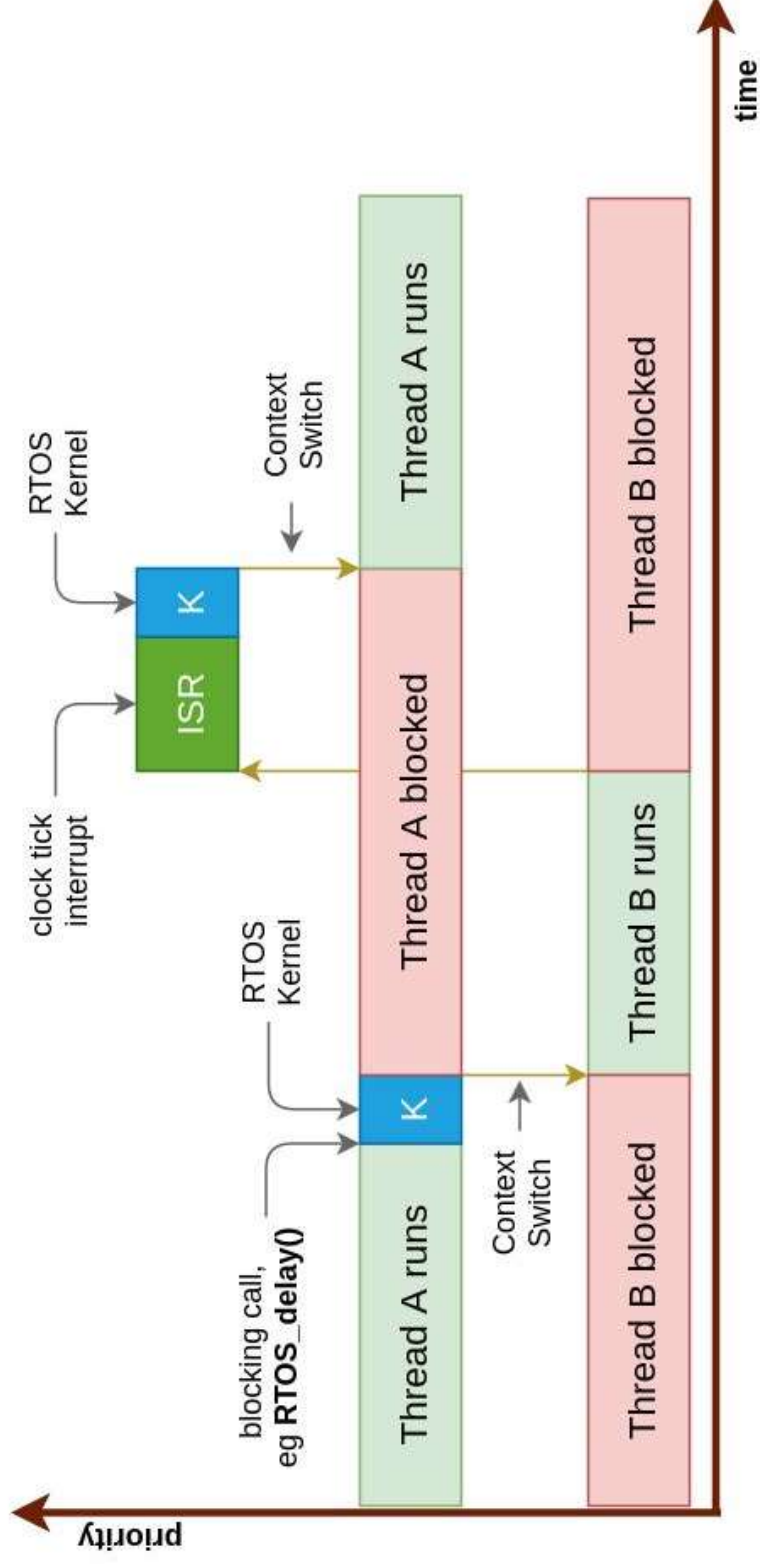
```
void thread_uart_tx()  
{  
    while (1)  
    {  
        if (uart_tx_ready())  
        {  
            c = uart_read();  
            queue.push(c);  
        }  
    }  
}
```

Sequential Programming with RTOS

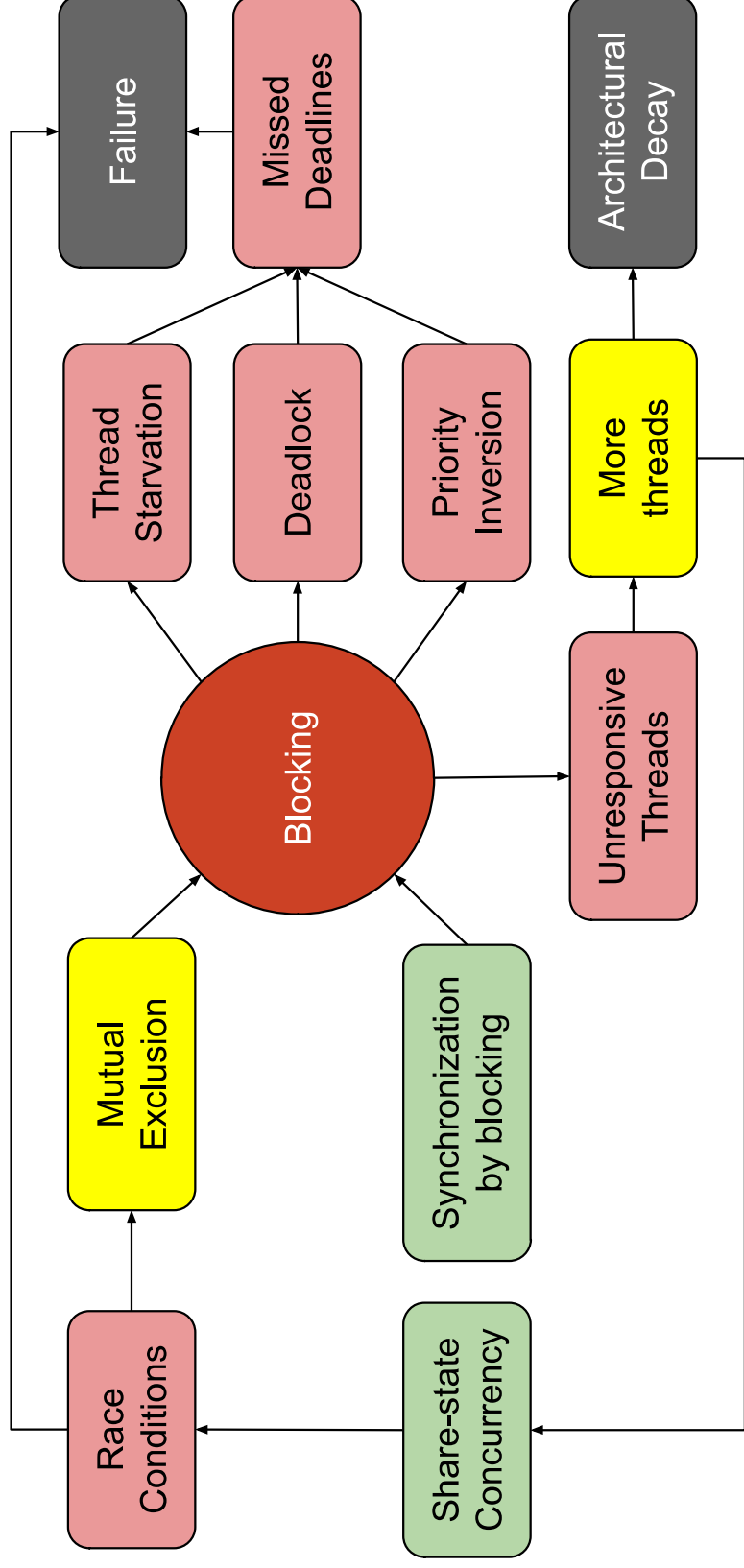
```
void thread_blink()  
{  
    pinMode(led, OUTPUT);  
    while (1)  
    {  
        digitalWrite(led, HIGH);  
        RTOS_delay(1000);  
        digitalWrite(led, LOW);  
        RTOS_delay(1000);  
    }  
}
```

```
void thread_uart_tx()  
{  
    while (1)  
    {  
        if (uart_tx_ready())  
        {  
            c = uart_read();  
            queue.push(c);  
        }  
    }  
}
```

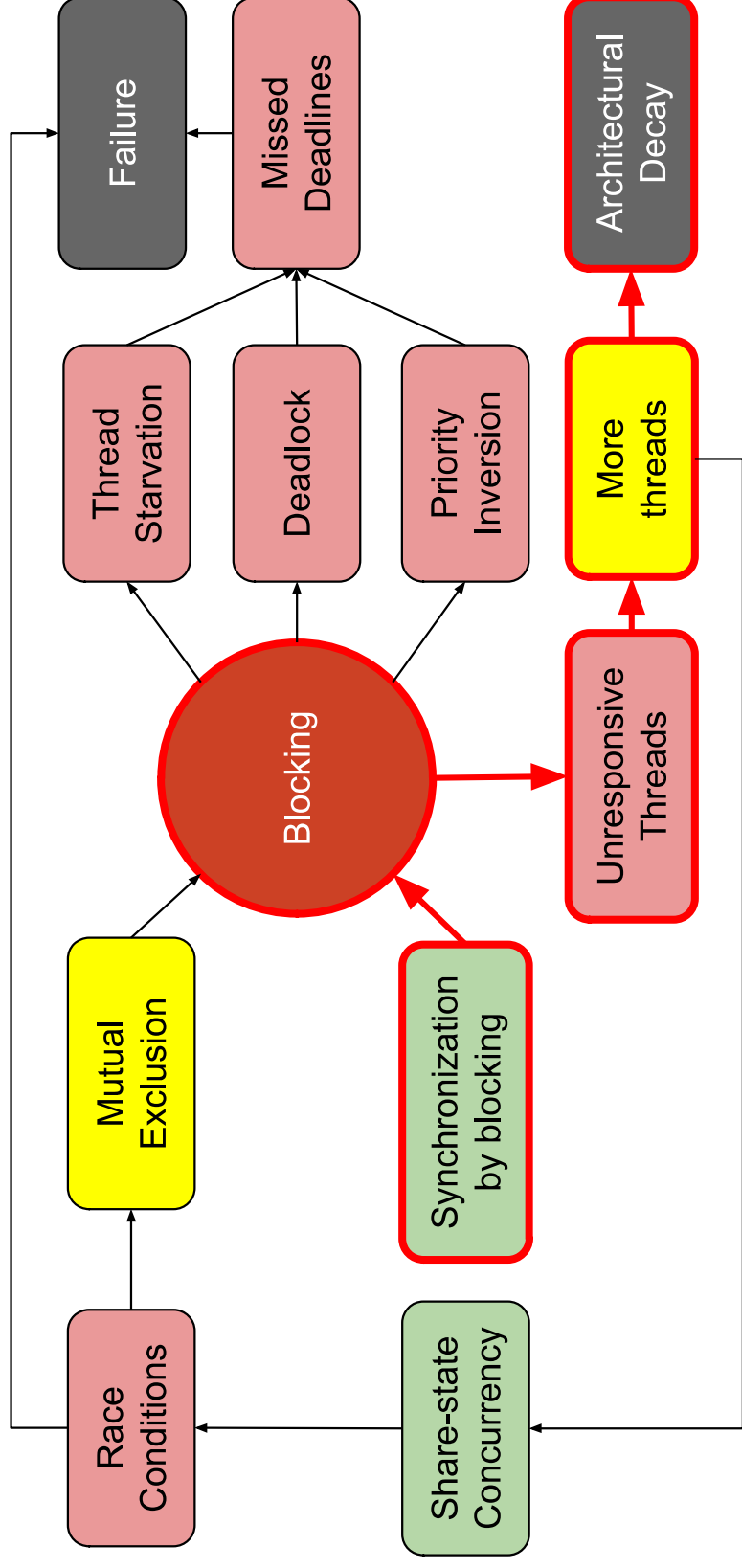
Sequential Programming with RTOS



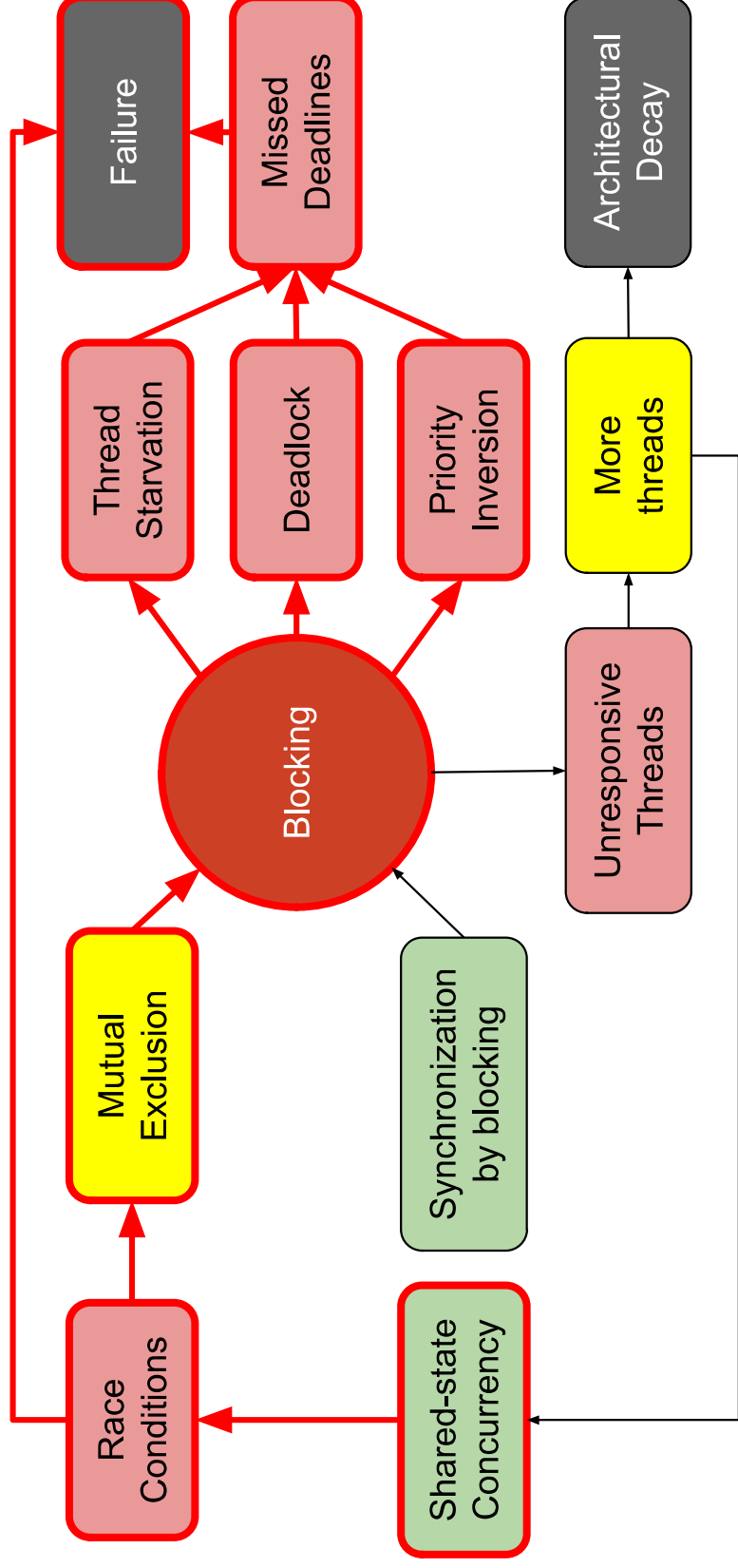
Problems of RTOS



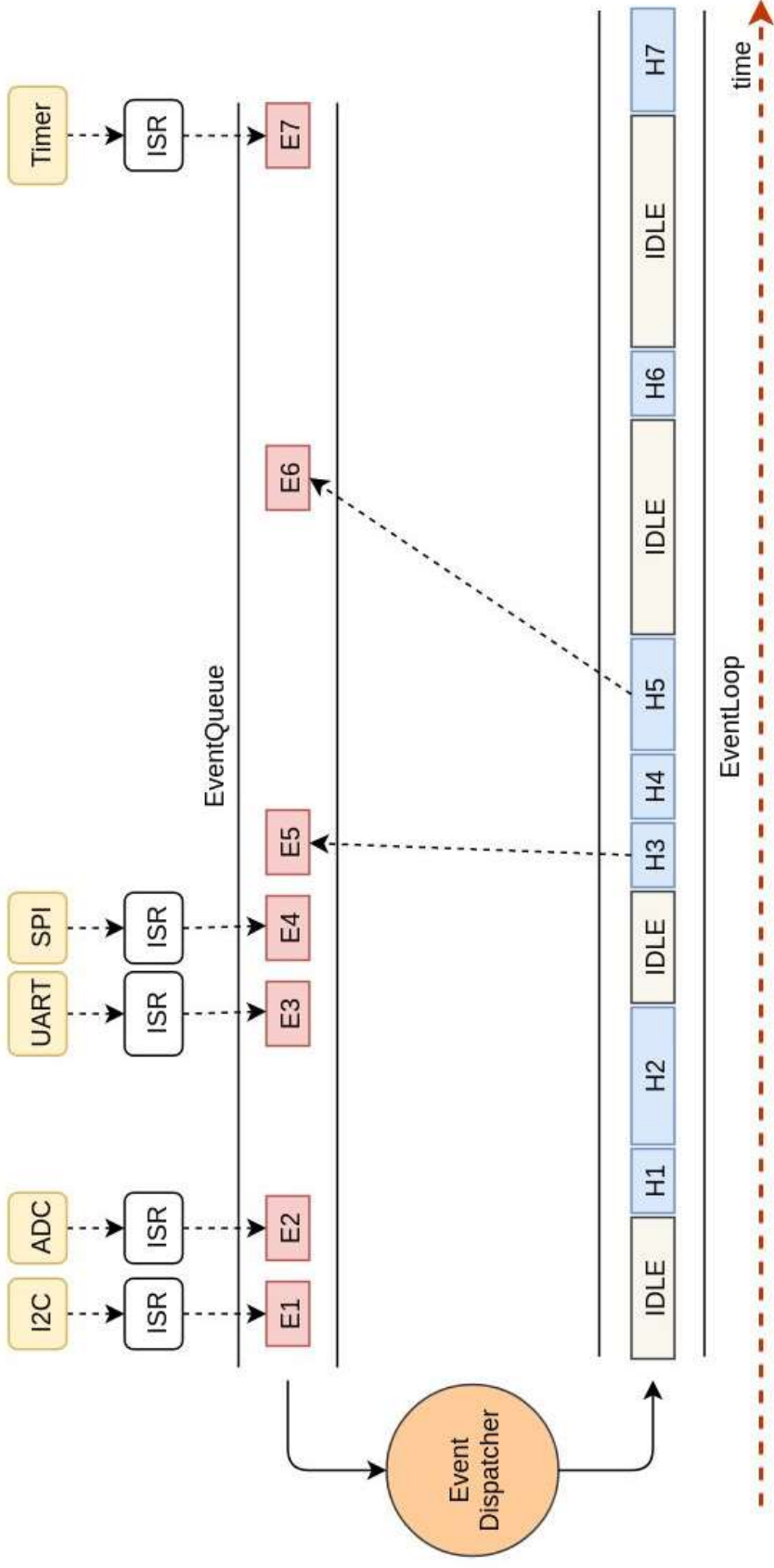
Problems of RTOS



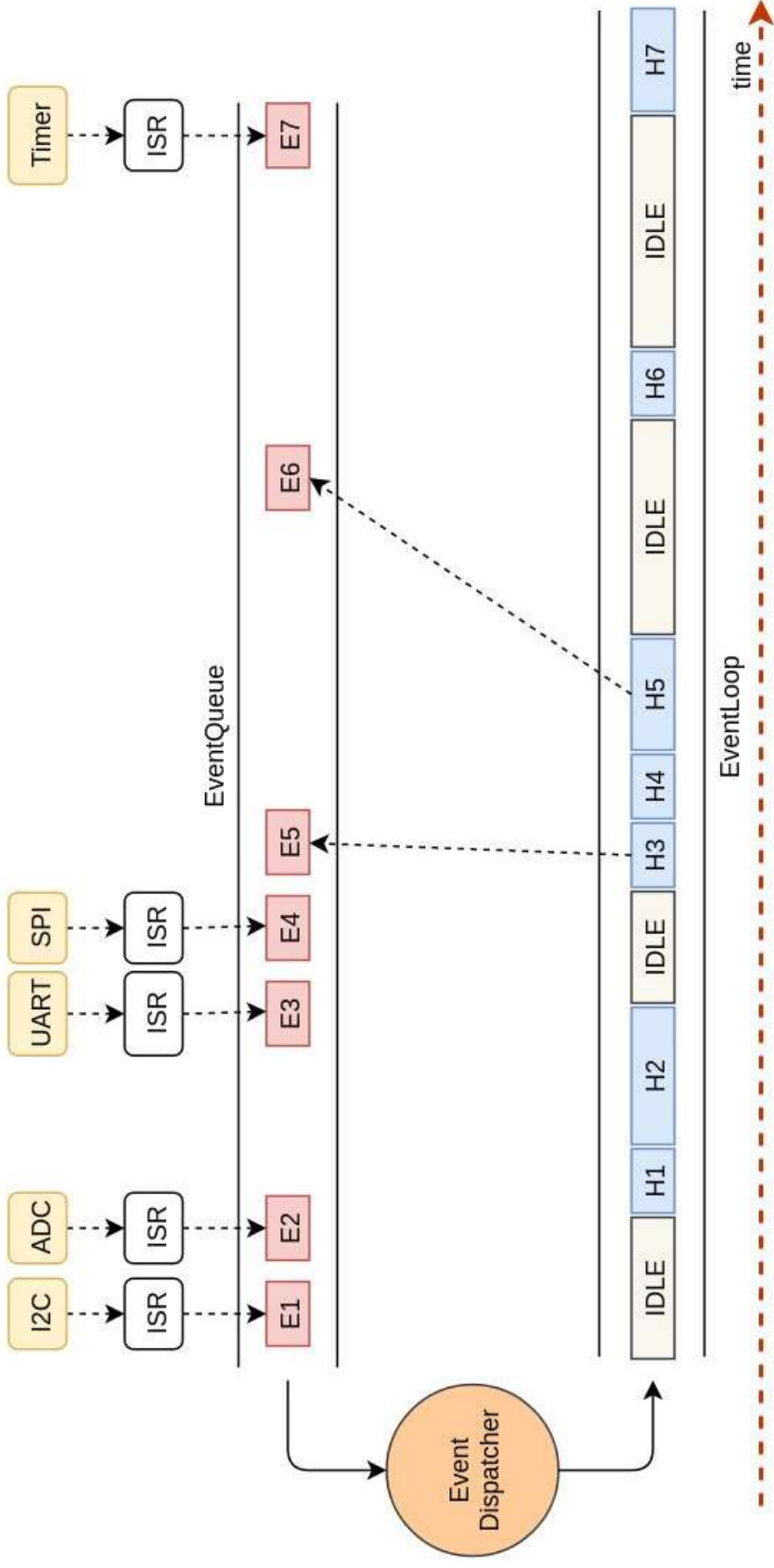
Problems of RTOS



Event-Driven Approach



Event-Driven Approach



Embedded Programming

Problems

- Use mainly sequential programming
- Lack of good code structure
- Spaghetti code everywhere
- Limited resources: CPU, RAM, STORAGE, POWER...
- Hard to debug and maintain

Solutions

- **Event-Driven Programming**
- Object Oriented Programming
- Finite State Machine
- Debug Tools

mStack

Embedded Programming

Problems

- Use mainly sequential programming
- Lack of good code structure
- Spaghetti code everywhere
- Limited resources: CPU, RAM, STORAGE, POWER...
- Hard to debug and maintain

Solutions

- Event-Driven Programming
- **Object Oriented Programming**
- Finite State Machine
- Debug Tools

mStack

Embedded Programming

Problems

- Use mainly sequential programming
- Lack of good code structure
- Spaghetti code everywhere
- Limited resources: CPU, RAM, STORAGE, POWER...
- Hard to debug and maintain

Solutions

- Event-Driven Programming
- Object Oriented Programming
- **Finite State Machine**
- Debug Tools

mStack

Embedded Programming

Problems

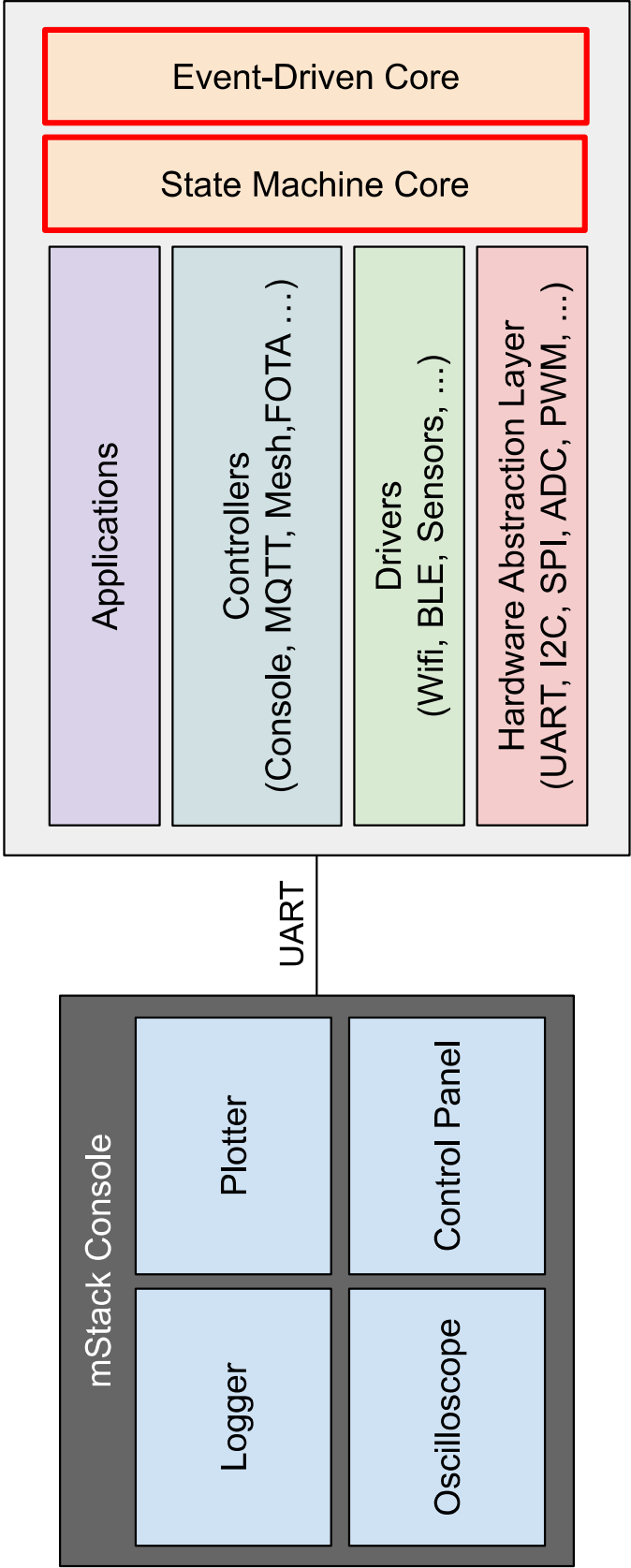
- Use mainly sequential programming
- Lack of good code structure
- Spaghetti code everywhere
- Limited resources: CPU, RAM, STORAGE, POWER...
- Hard to debug and maintain

Solutions

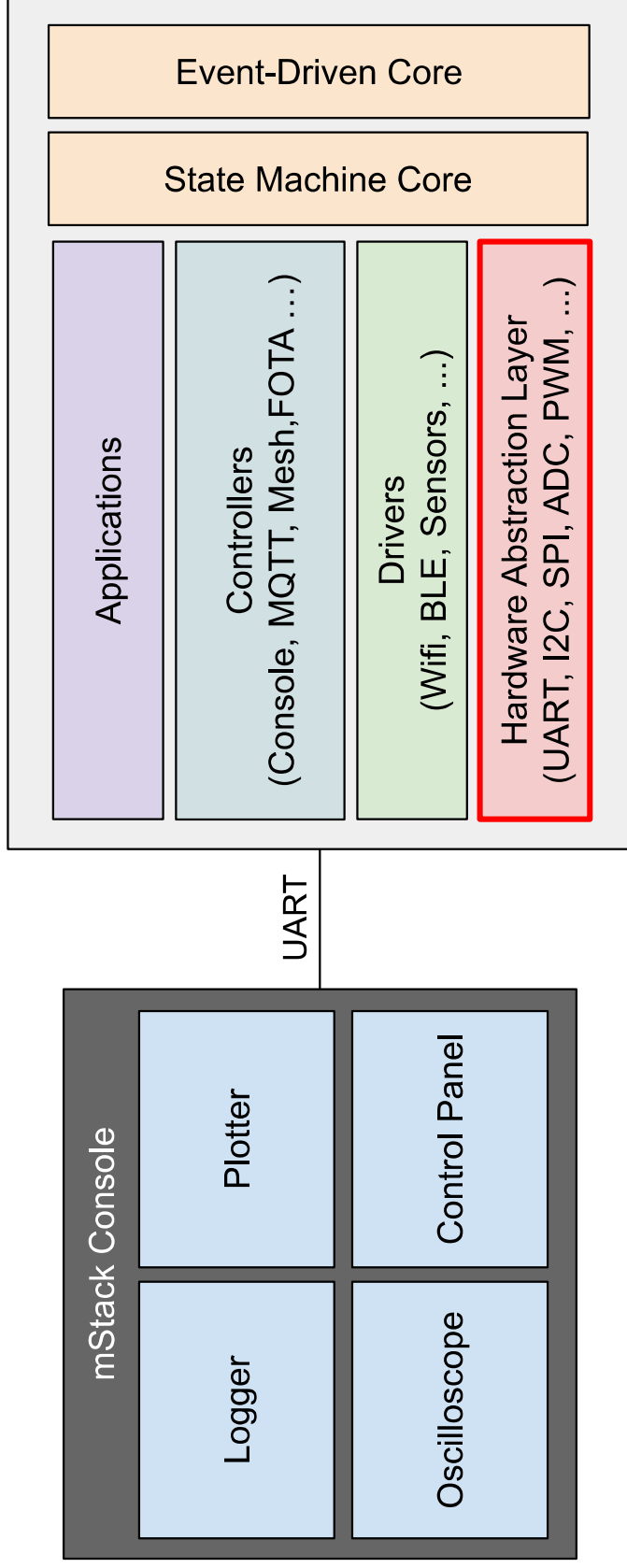
- Event-Driven Programming
- Object Oriented Programming
- Finite State Machine
- **Debug Tools**

mStack

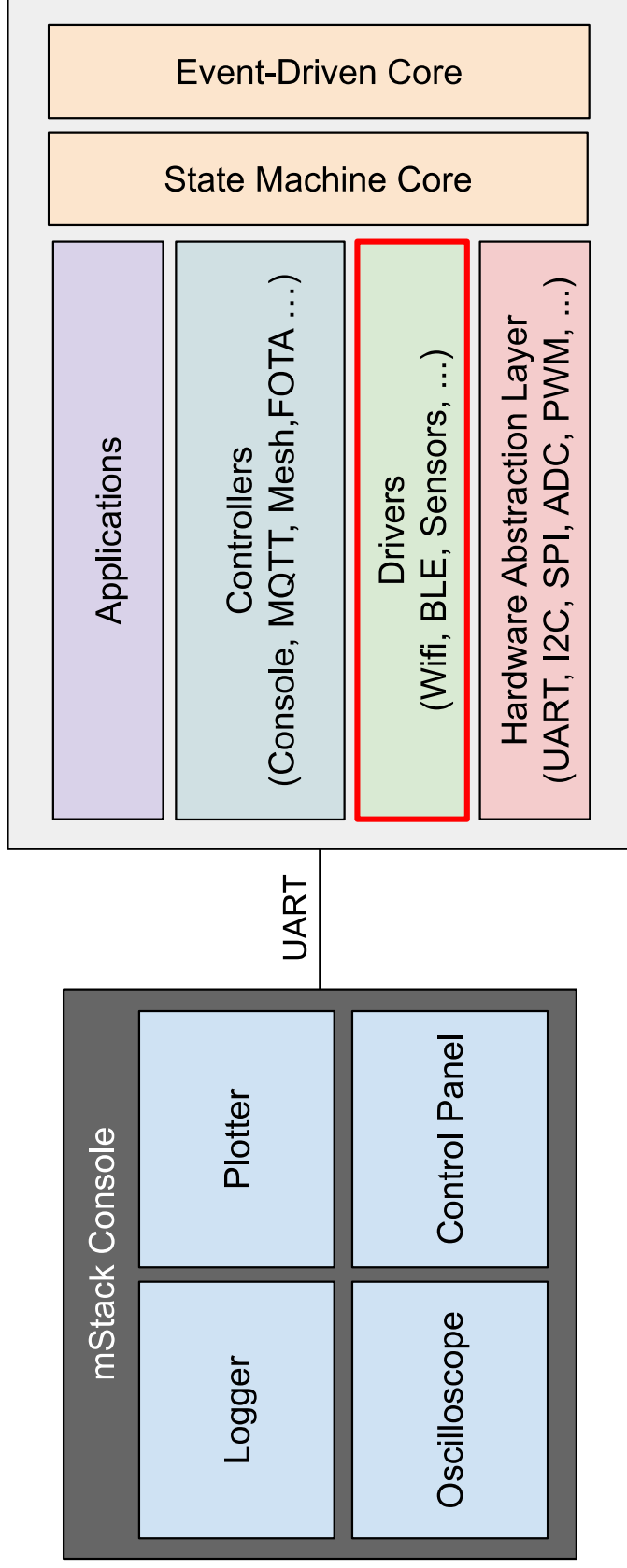
Architecture



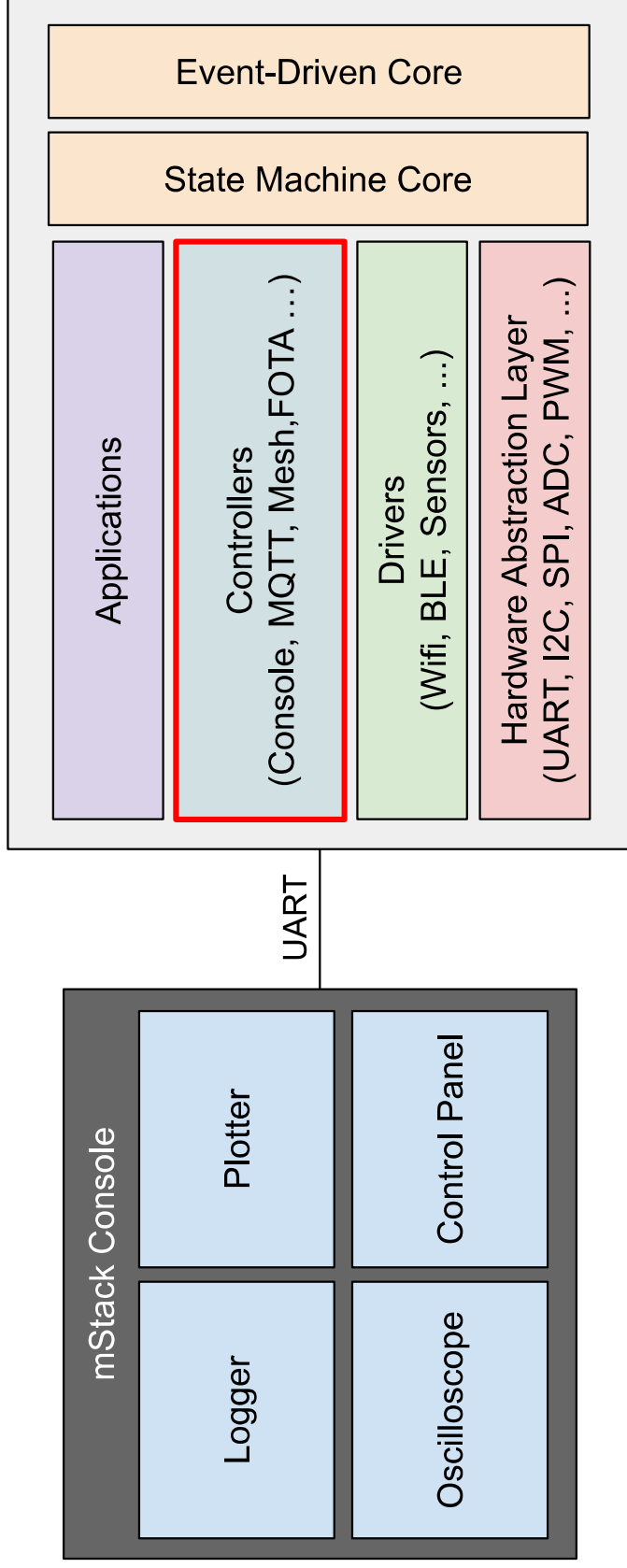
Architecture



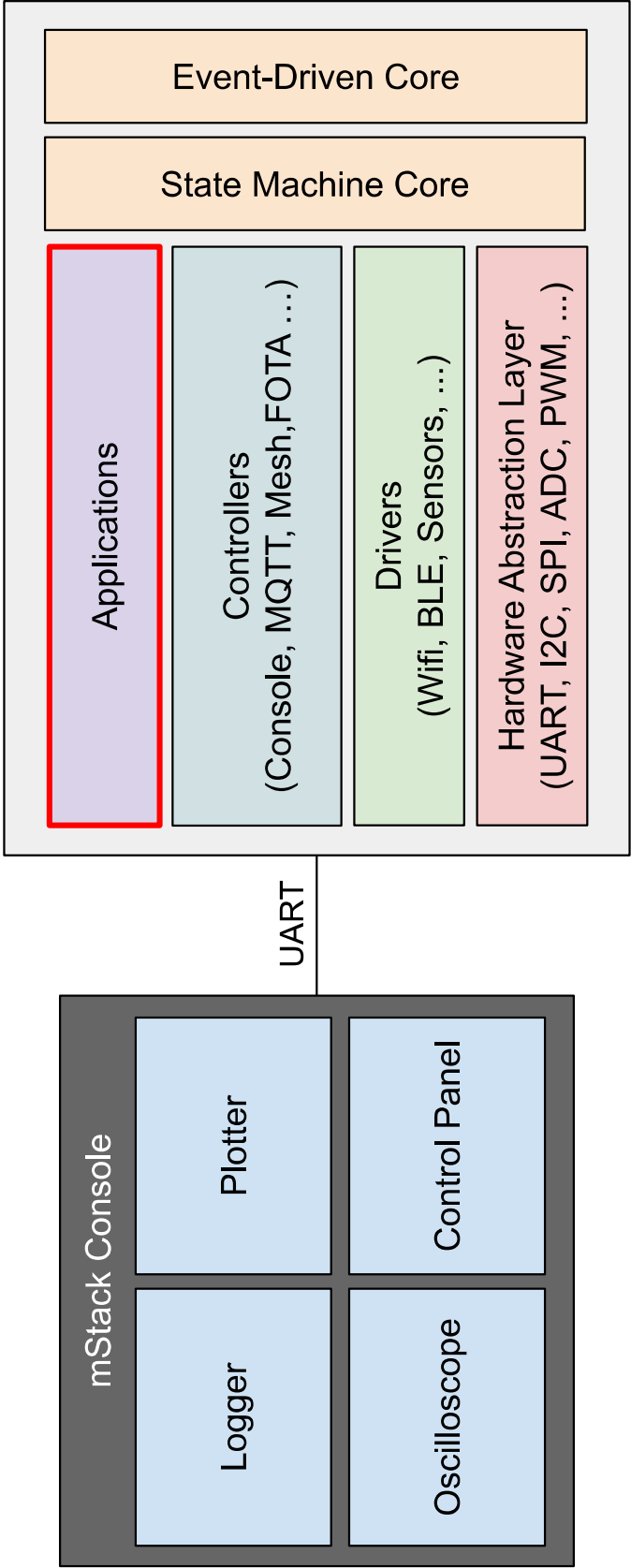
Architecture



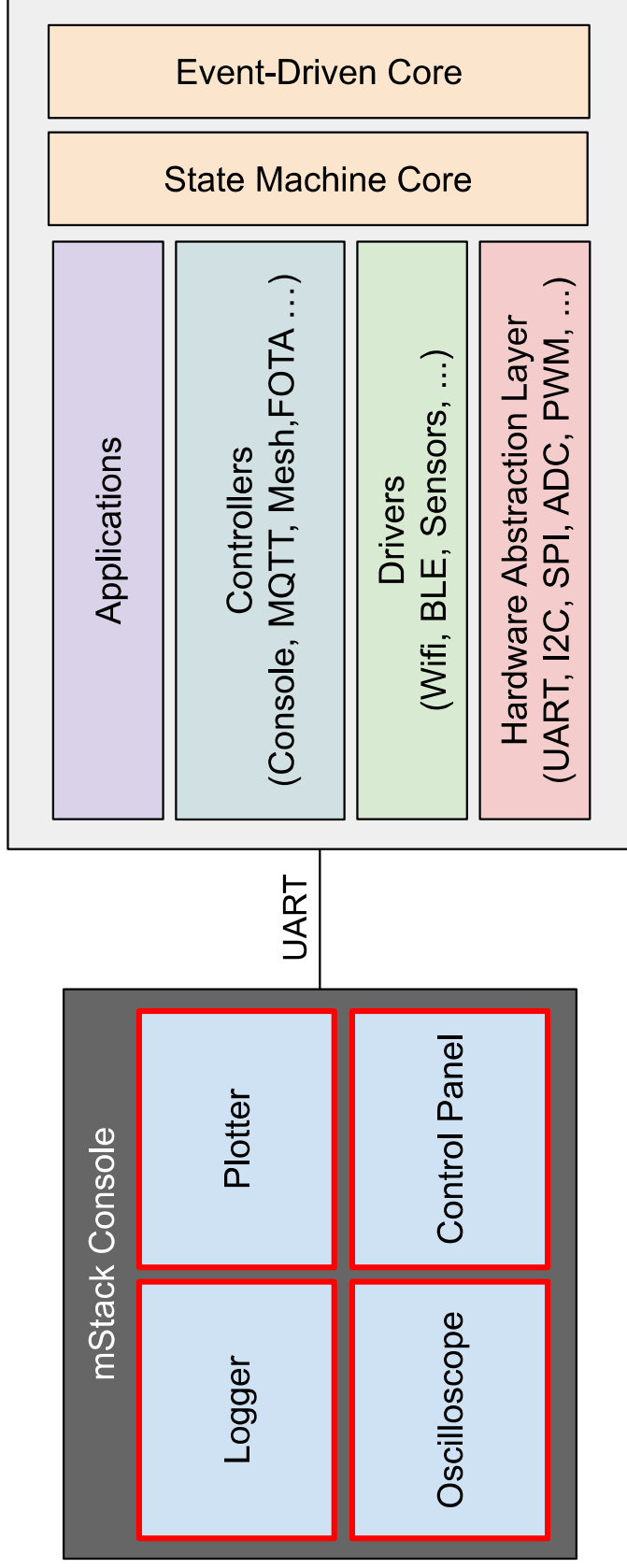
Architecture



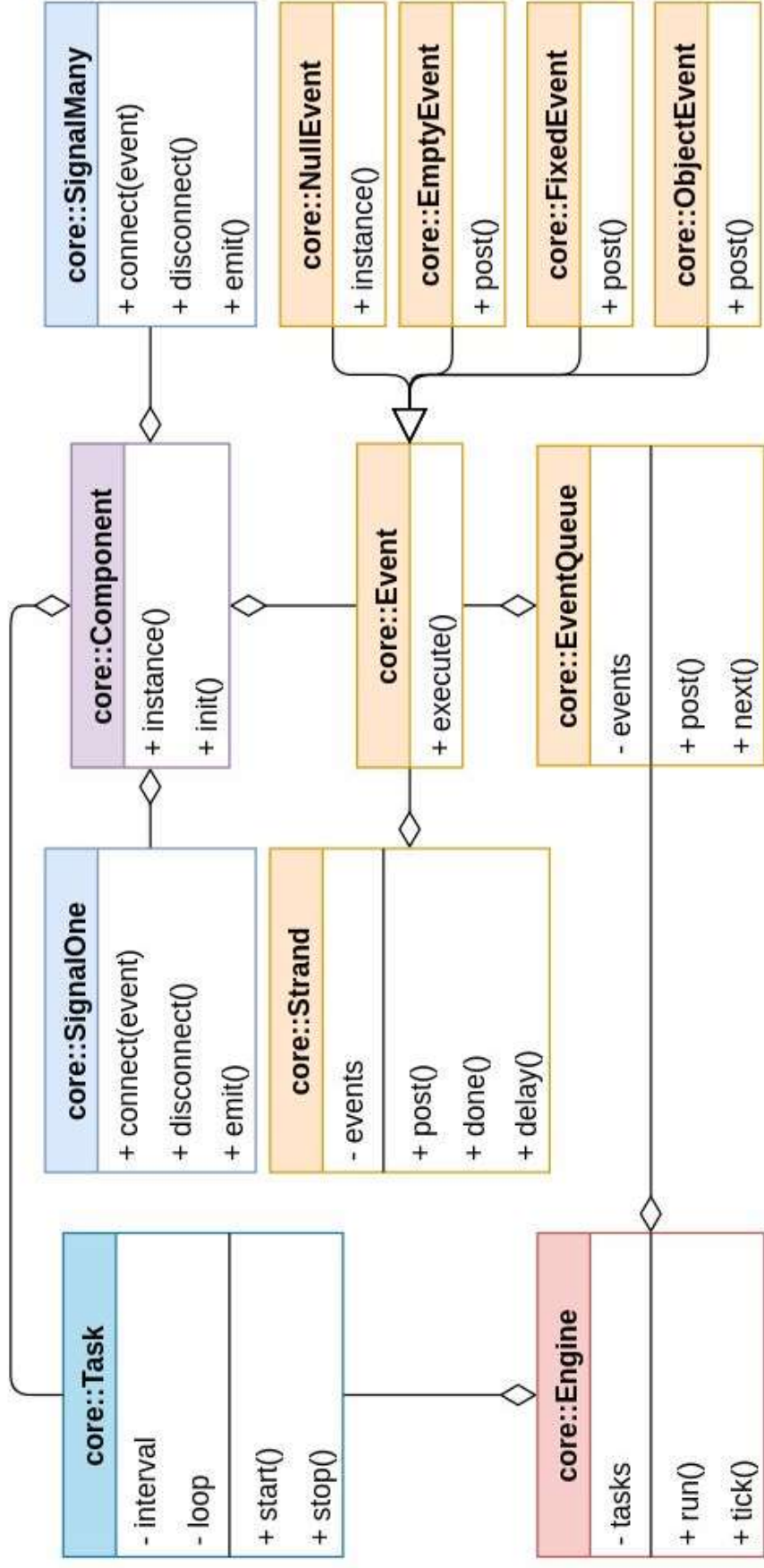
Architecture



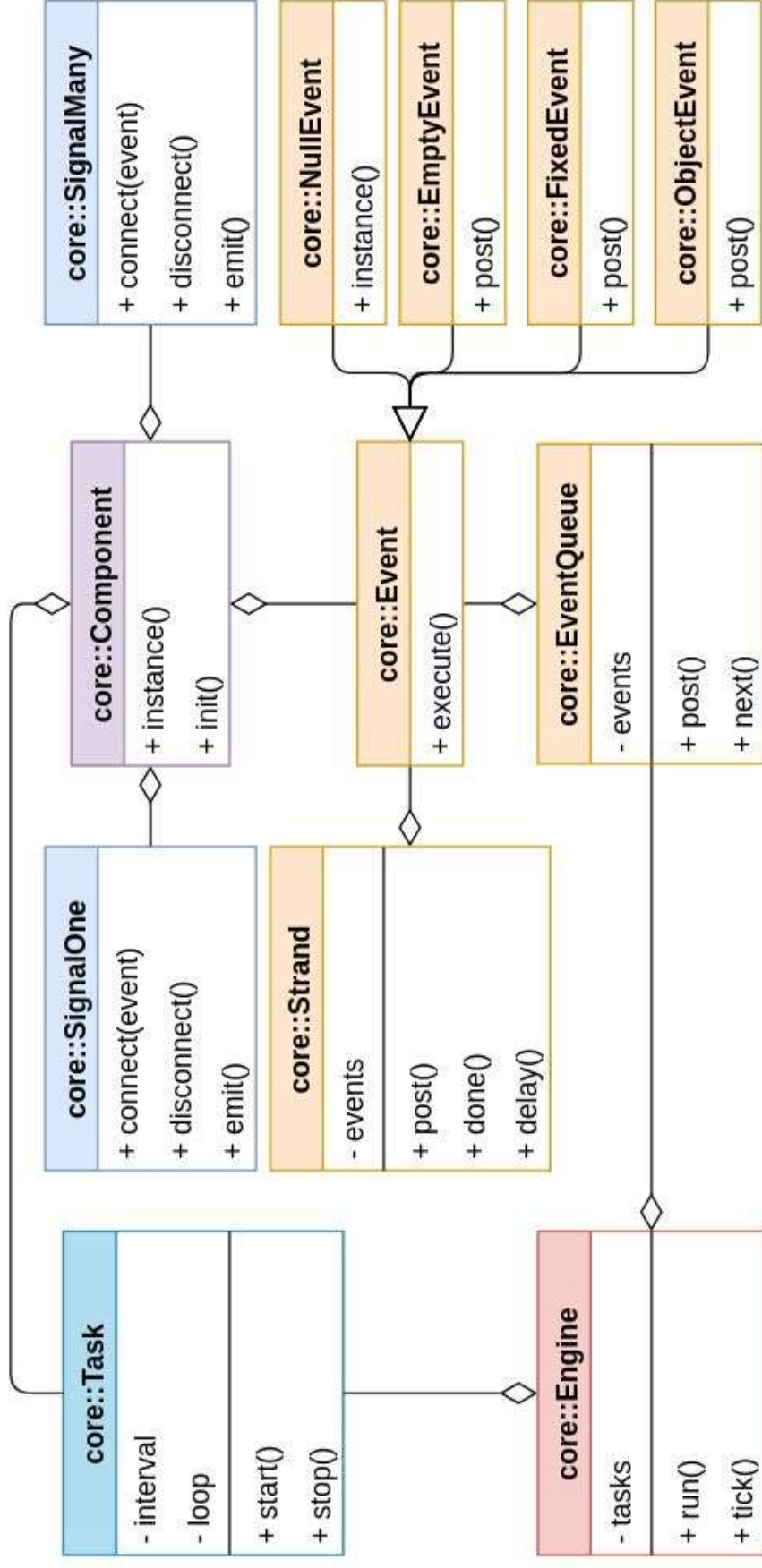
Architecture



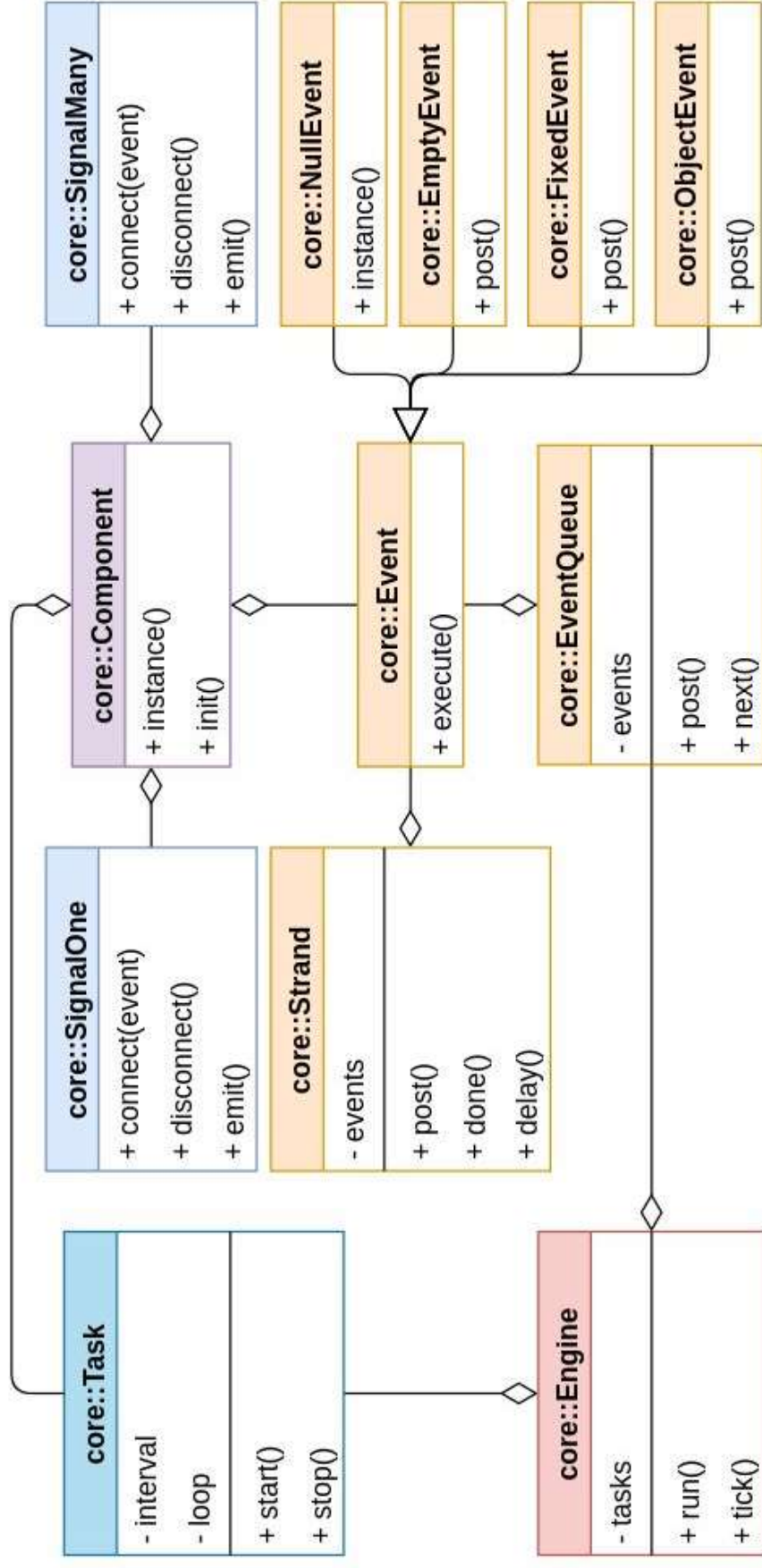
Event-Driven Core



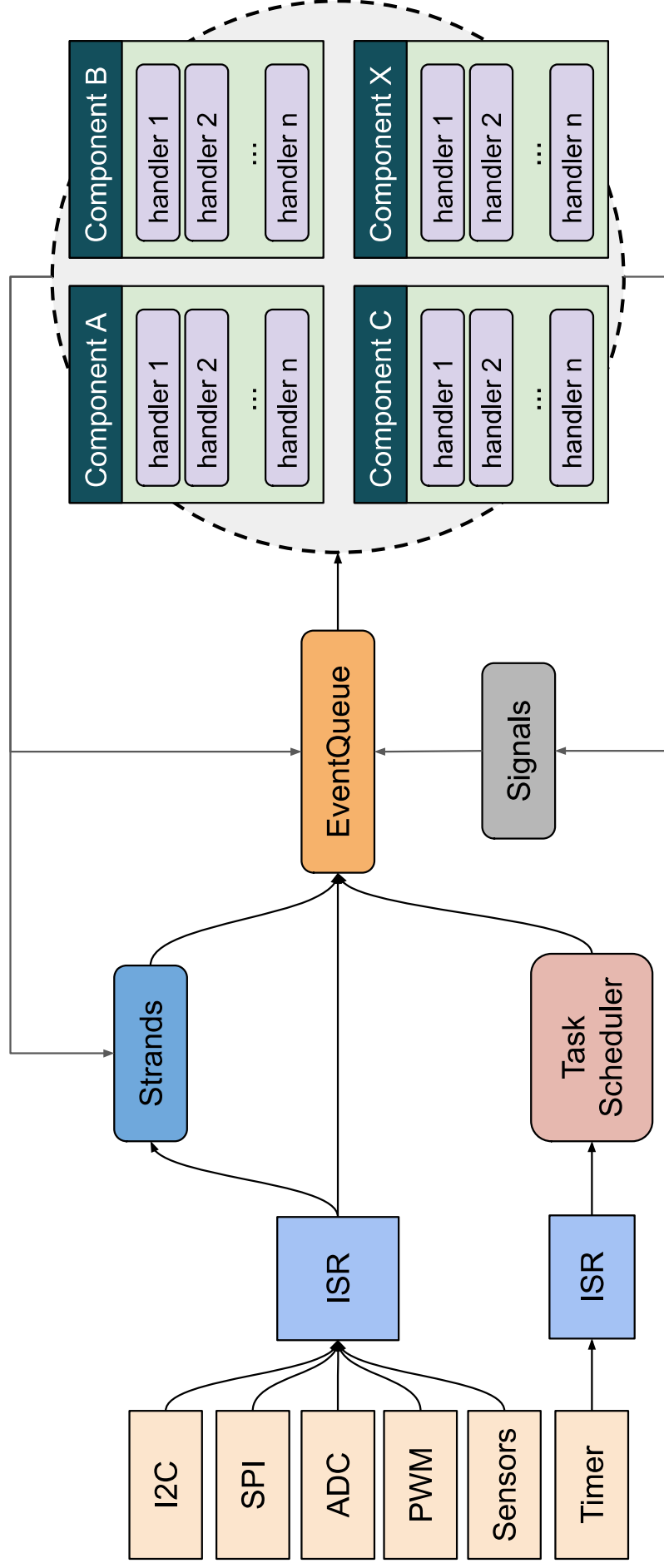
Event-Driven Core



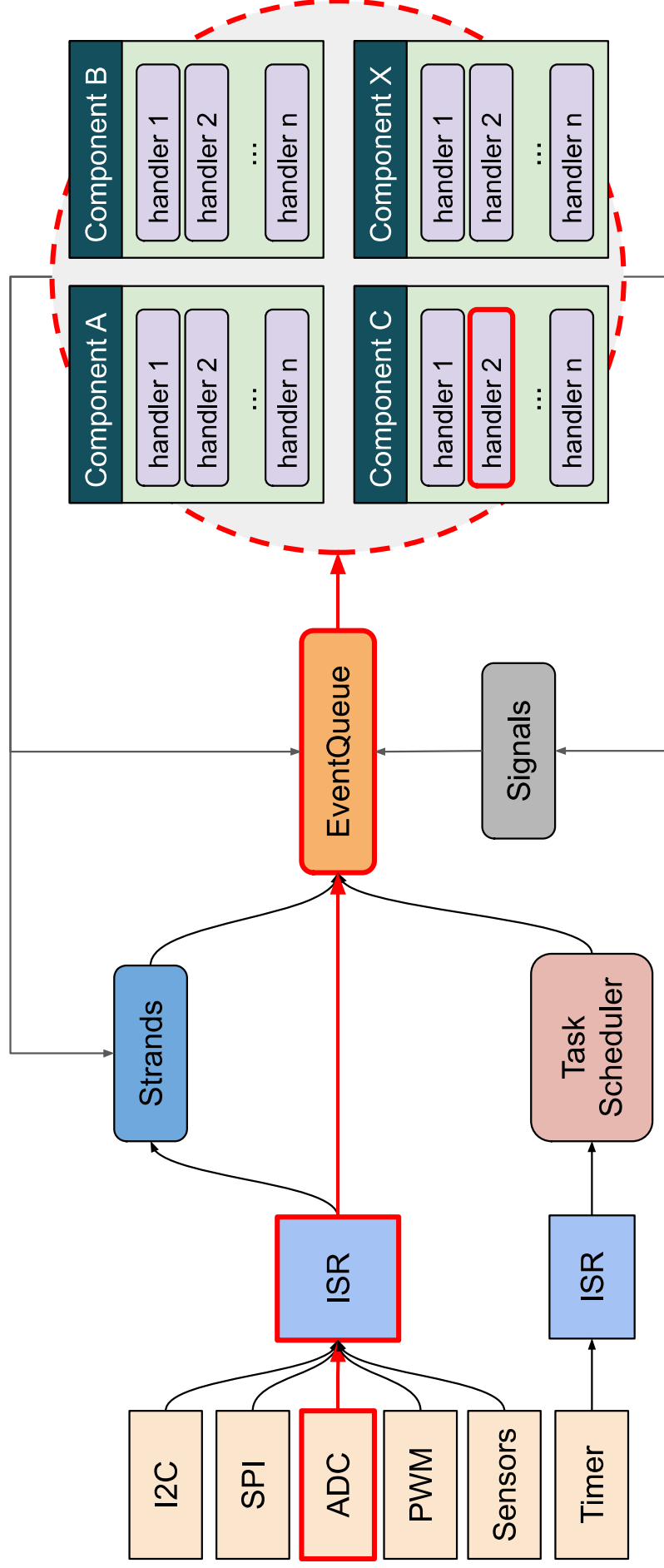
Event-Driven Core



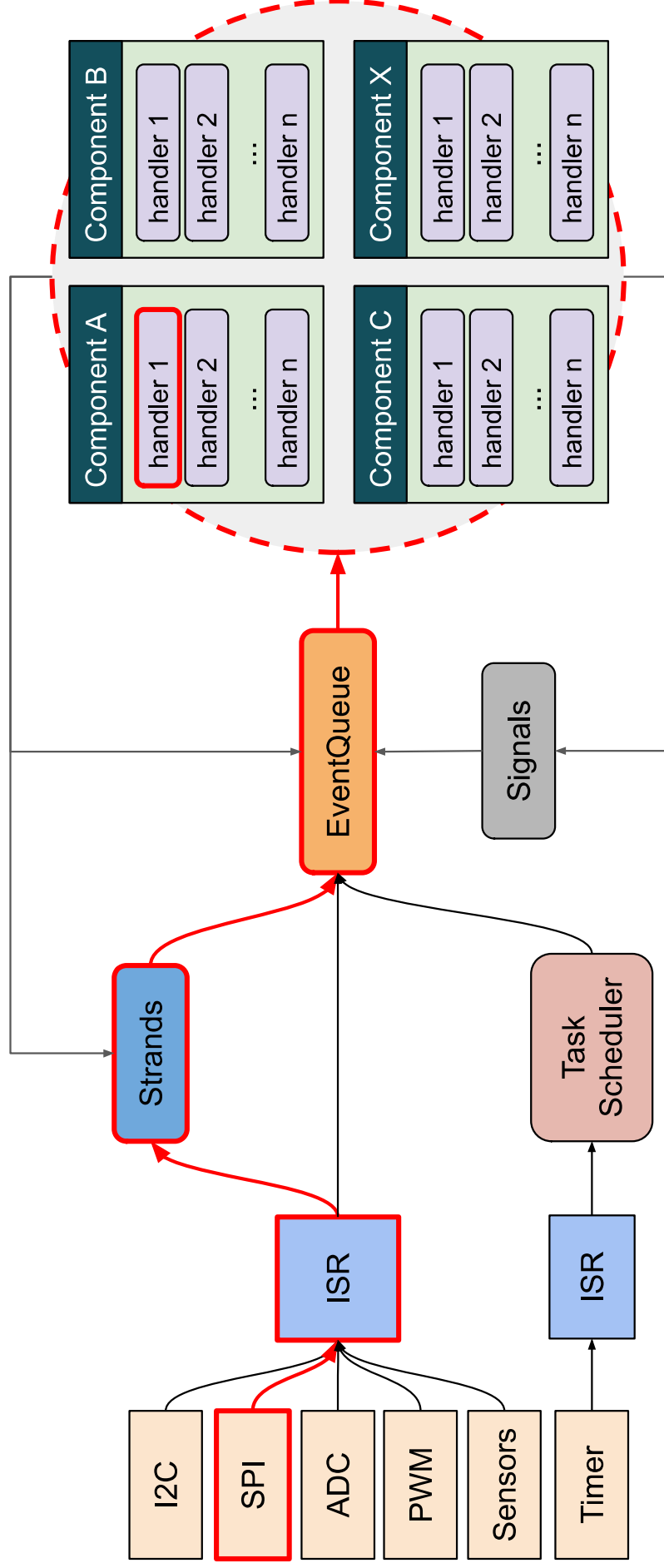
Event-Driven Core



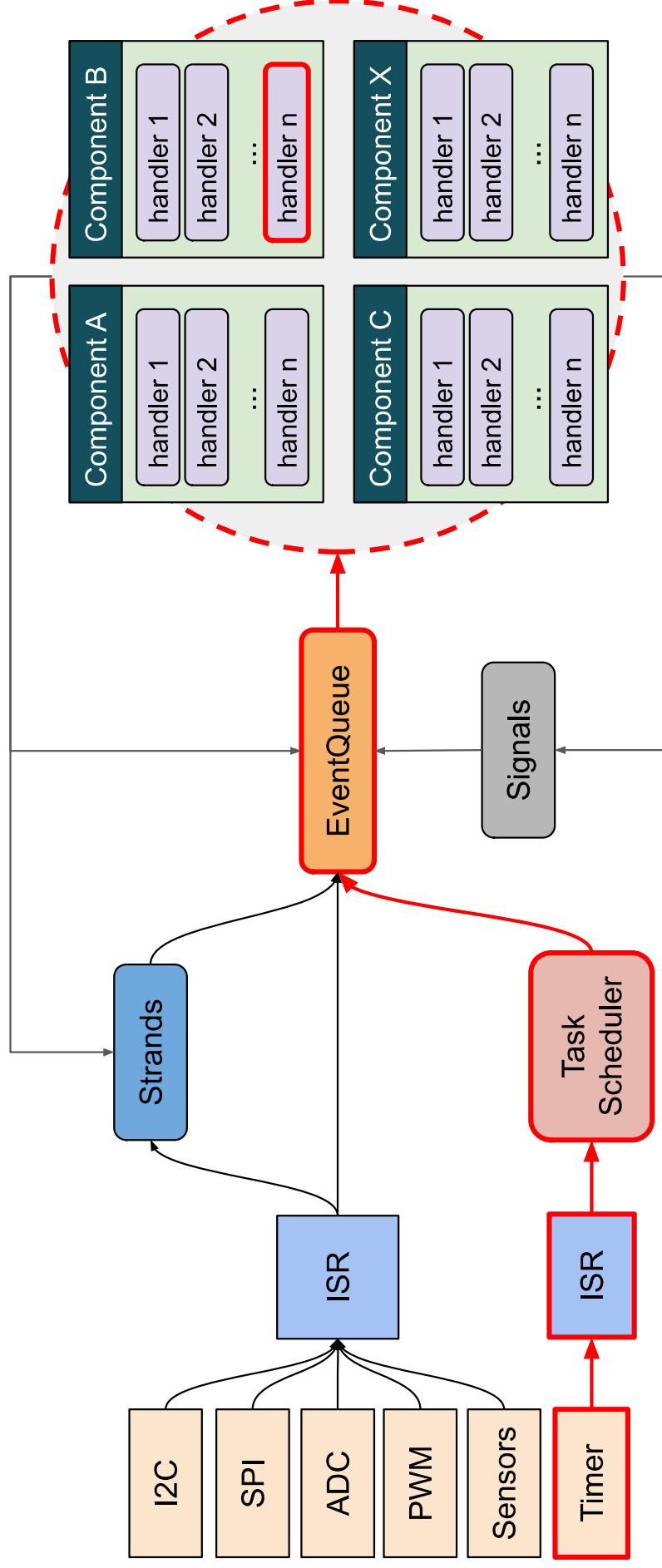
Event-Driven Core



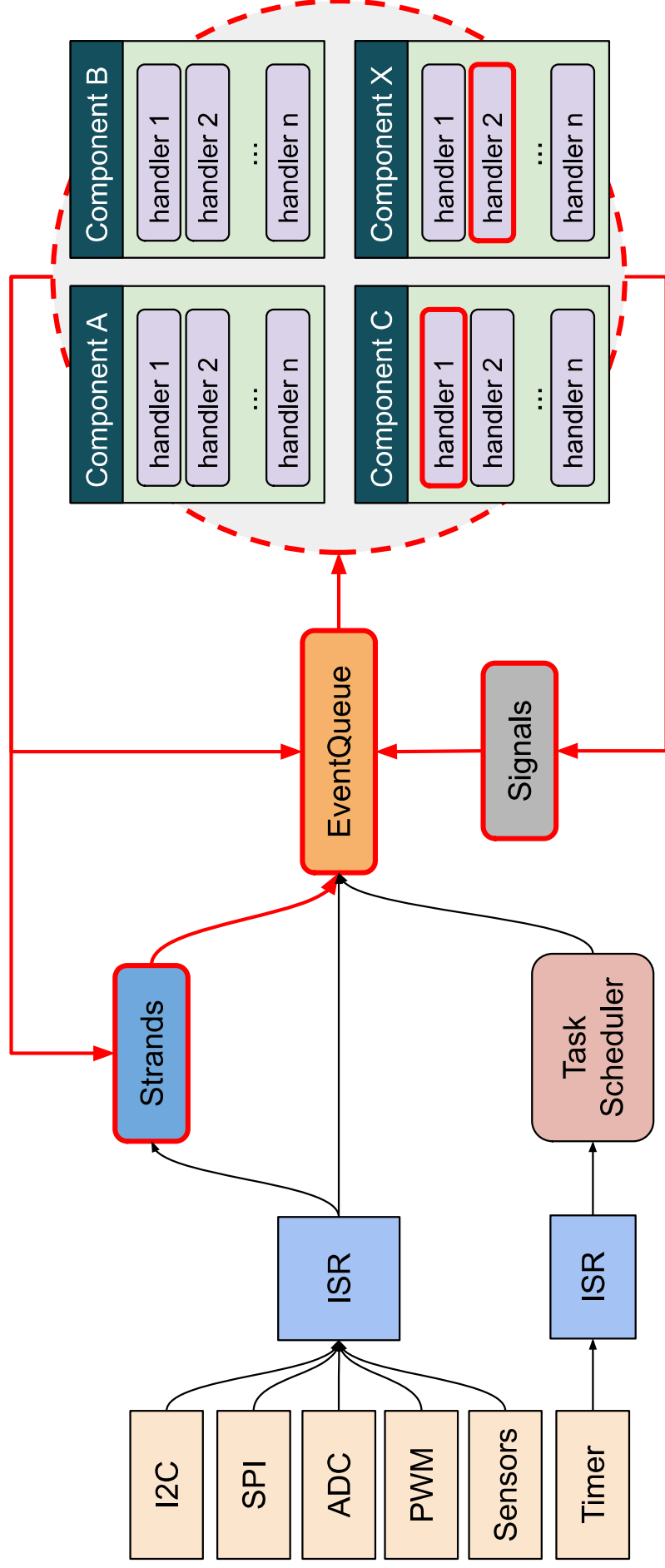
Event-Driven Core



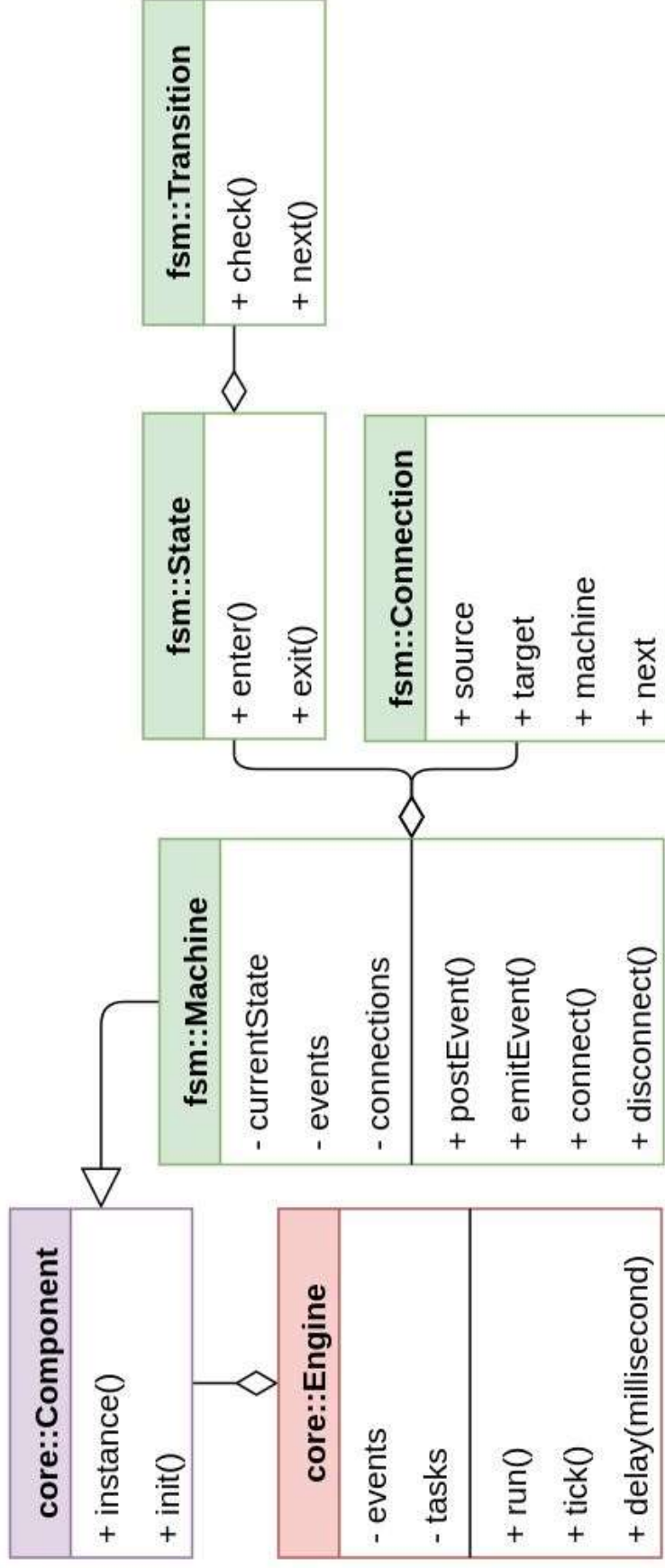
Event-Driven Core



Event-Driven Core



State Machine Core



Console



Conclusion

- Sequential and Event-Driven Programming
- mStack
- Next Talks
 - Event-Driven Programming with mStack
 - State Machine Programming with mStack
 - Using mStack Console