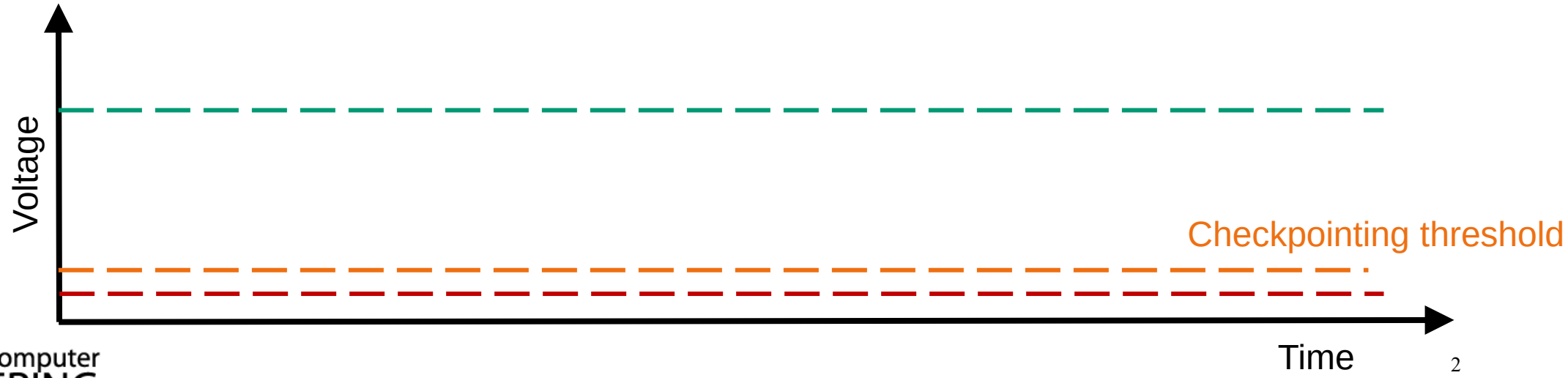
System Challenges of Intermittent Computing

Kiwan Maeng

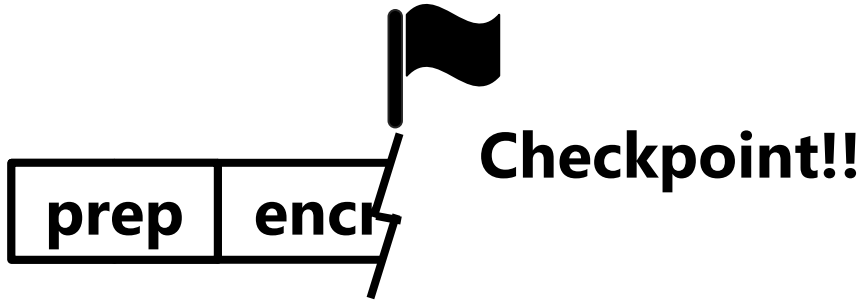
Recap: Just-In-Time (JIT) Checkpointing Enables Intermittent Execution

Workload

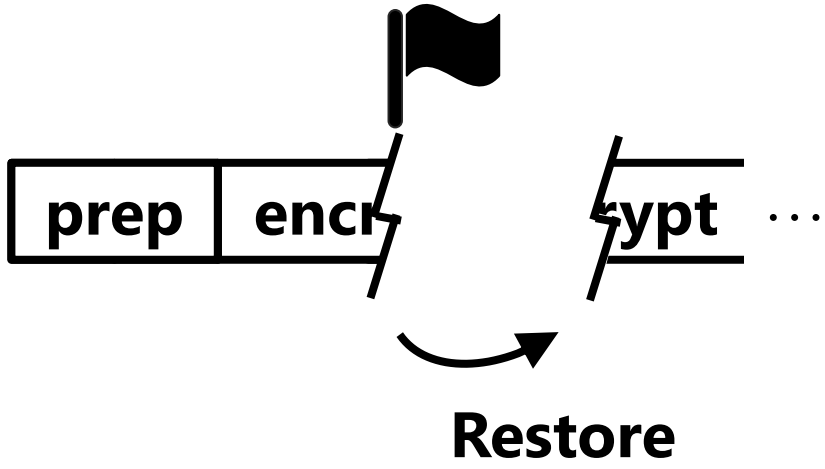


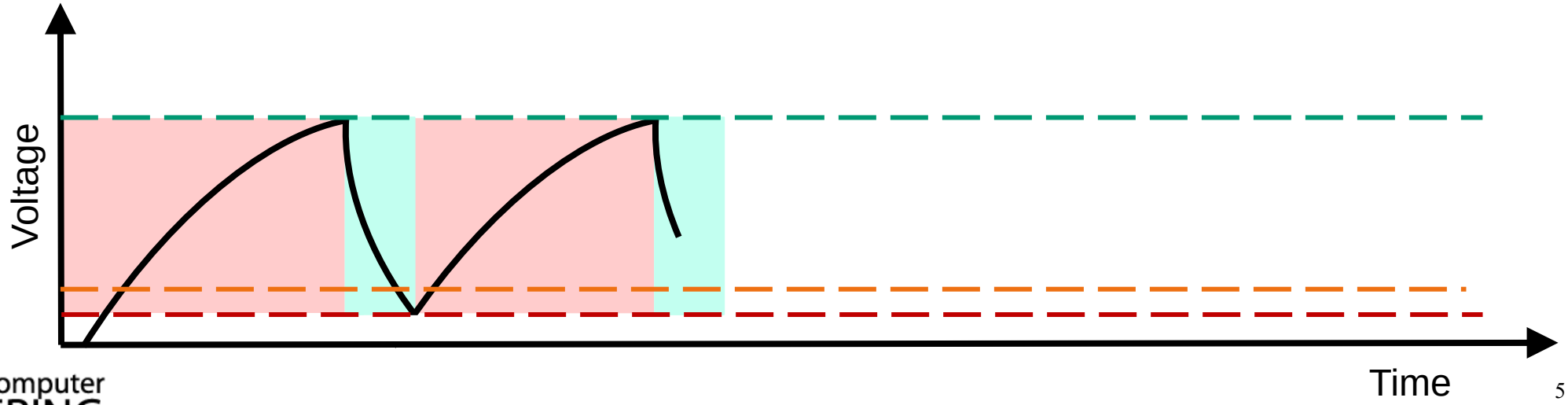
Recap: JIT Checkpointing Enables Intermittent Execution

Recap: JIT Checkpointing Enables Intermittent Execution

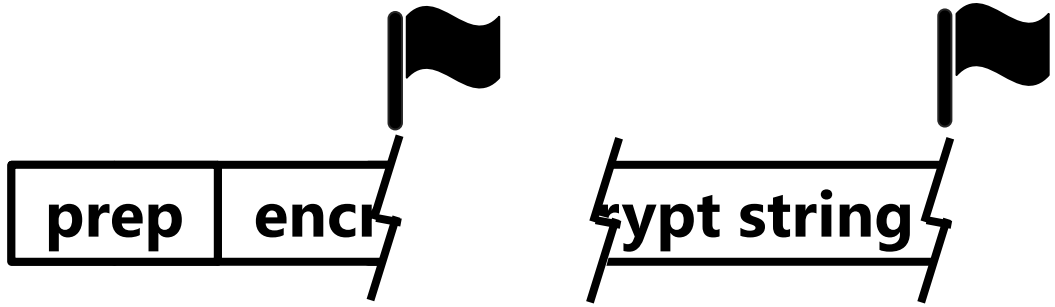


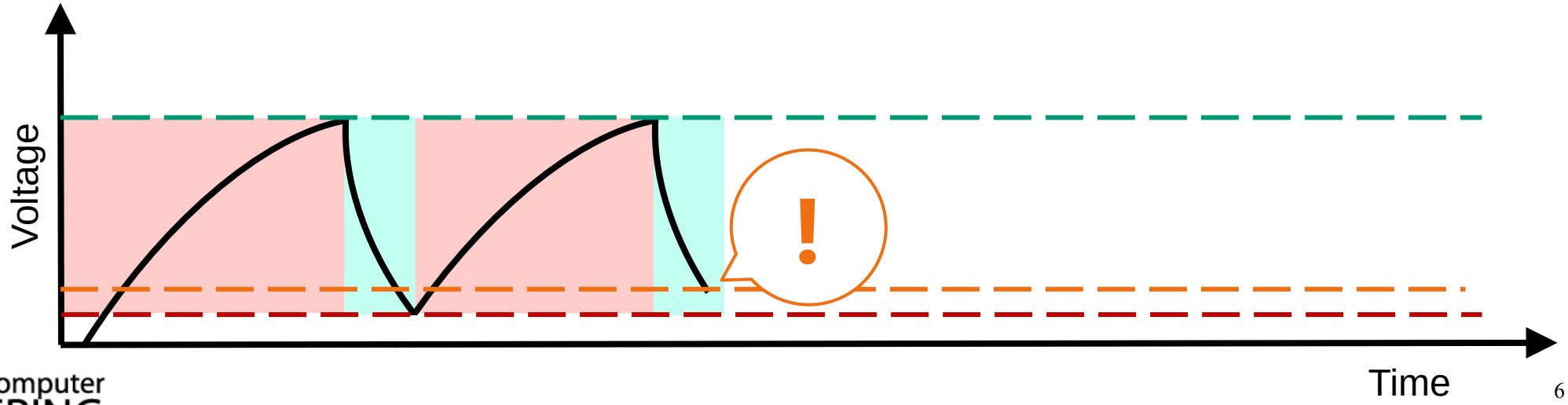
Recap: JIT Checkpointing Enables Intermittent Execution



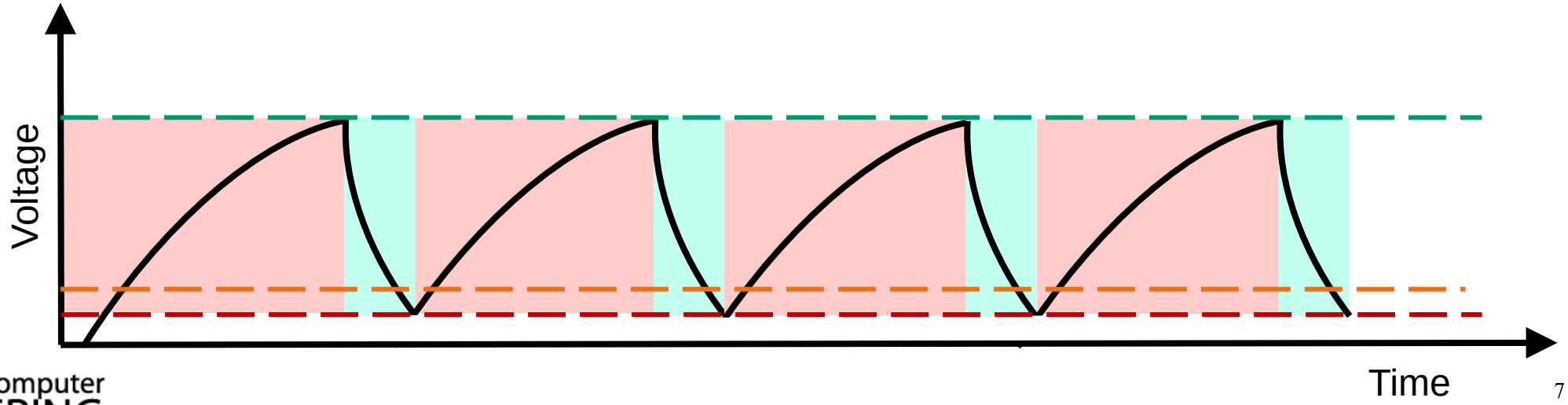


Recap: JIT Checkpointing Enables Intermittent Execution



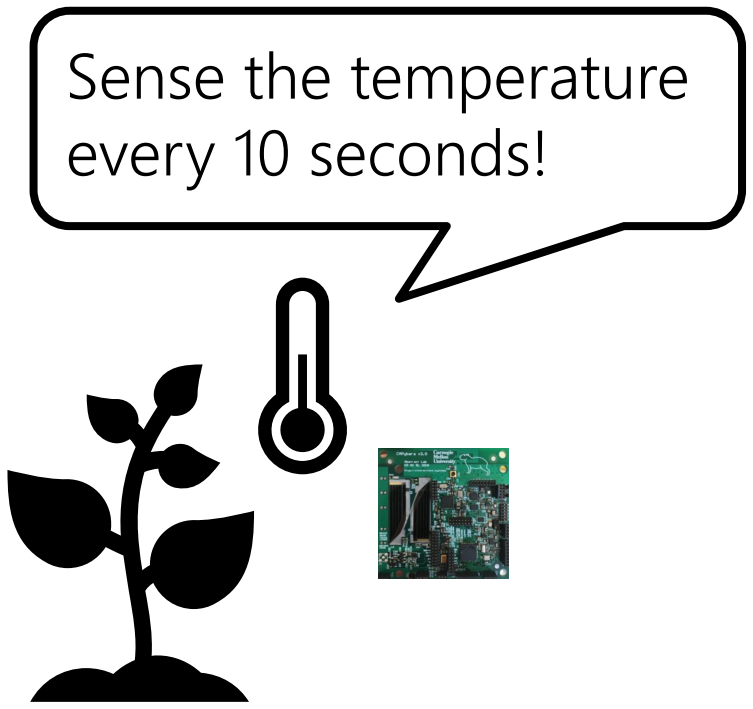


Recap: JIT Checkpointing Enables Intermittent Execution



Real-world Applications Need More Than Computation

Periodic Execution



Smart agriculture

[Vasisht 2017]

Atomic Execution

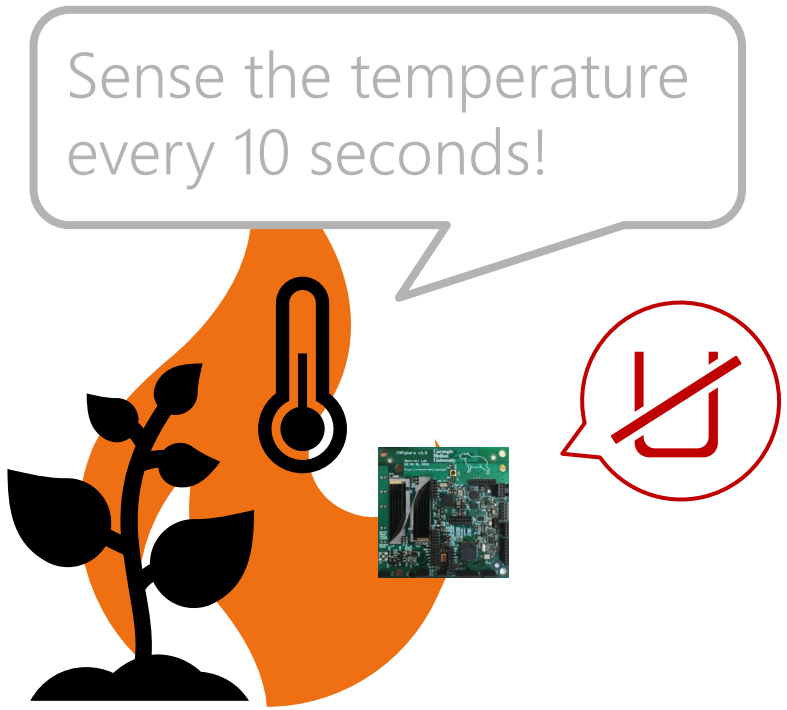
Wildlife tracking

[Zhang 2004]

Electrical & Computer ENGINEERING

Intermittence Complicates Additional Execution Models

Periodic Execution

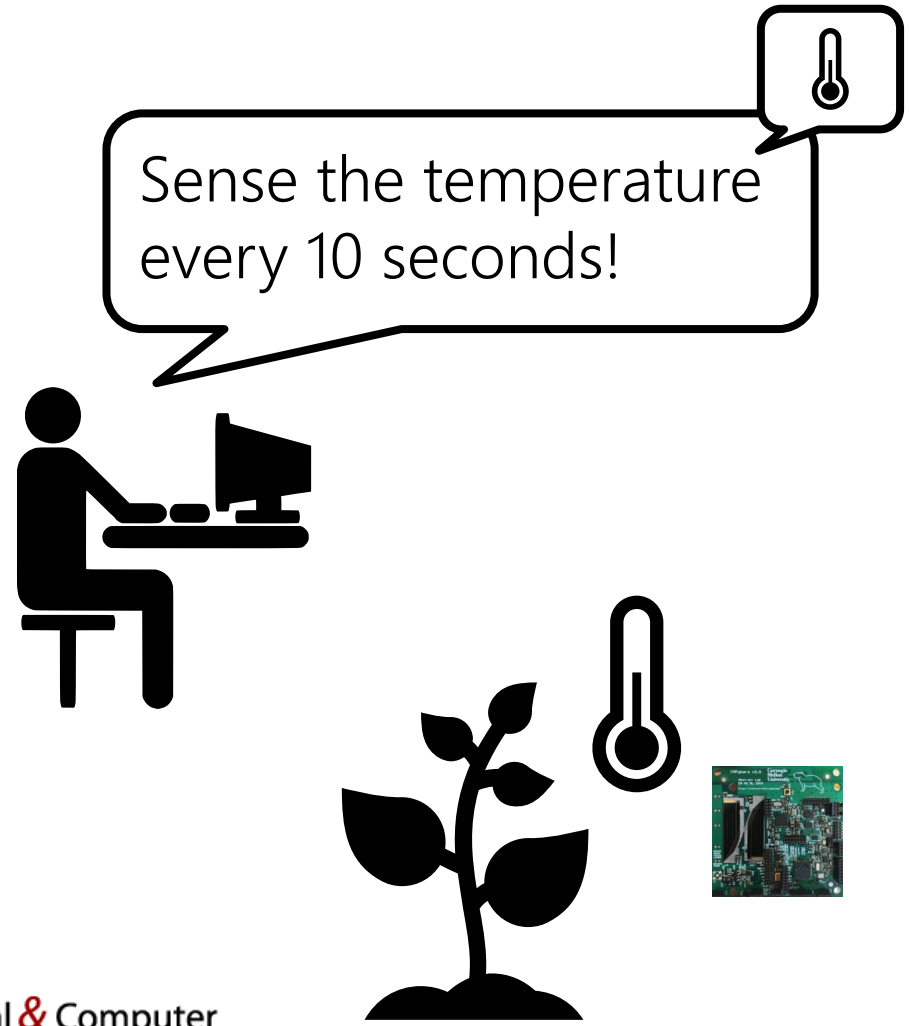


Atomic Execution

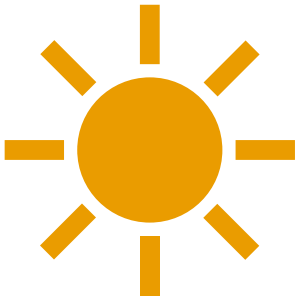
Outline

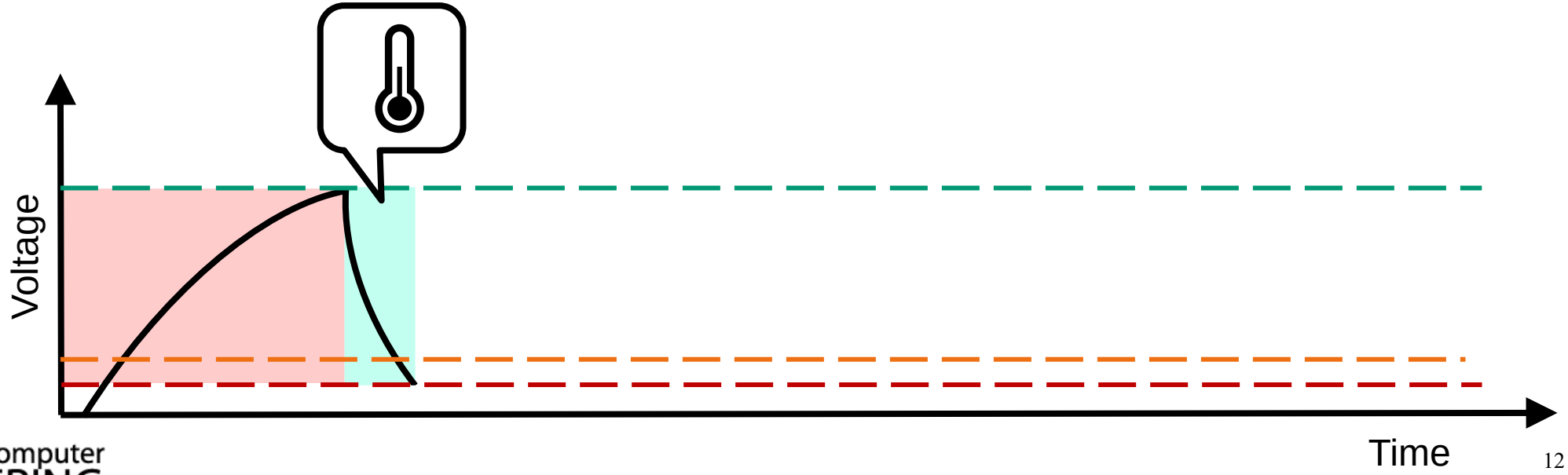
- **Challenge 1. Periodic Execution**
- Solution 1. CatNap
- Challenge 2. Atomic Execution
- Solution 2. Samoyed
- Future Work

Power Failures Complicate Periodic Execution

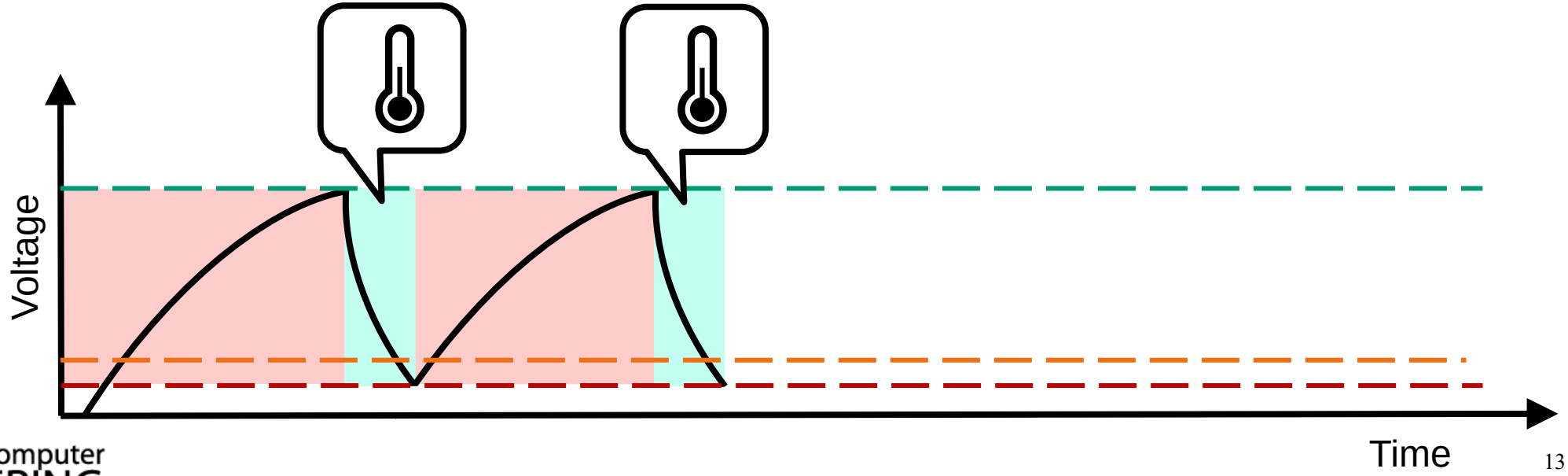


Power Failures Complicate Periodic Execution

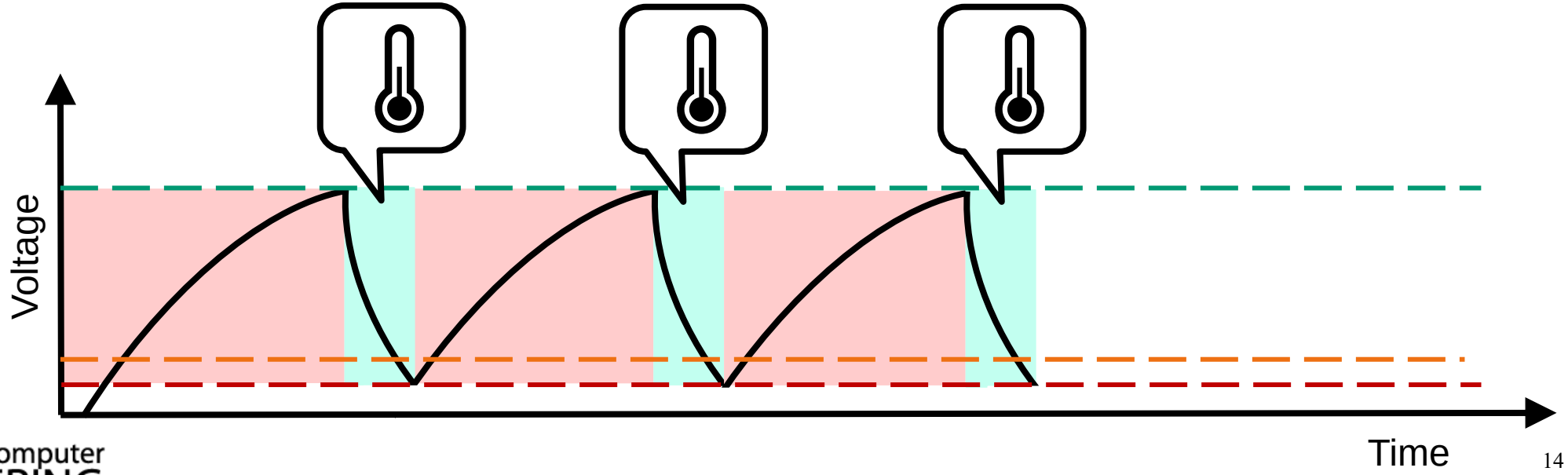




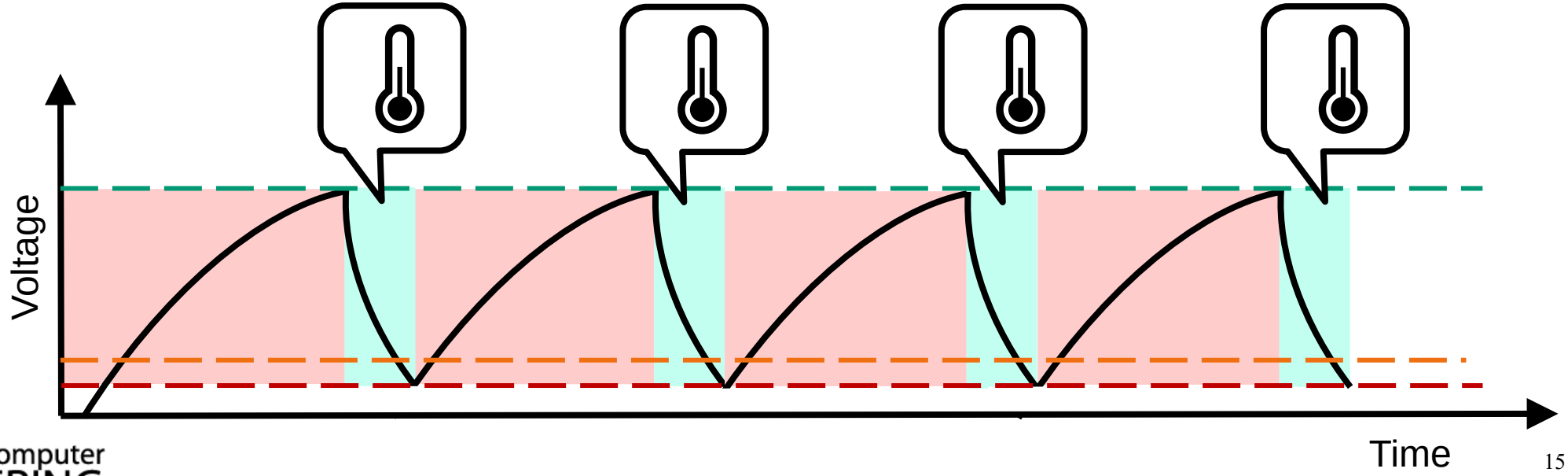
Power Failures Complicate Periodic Execution



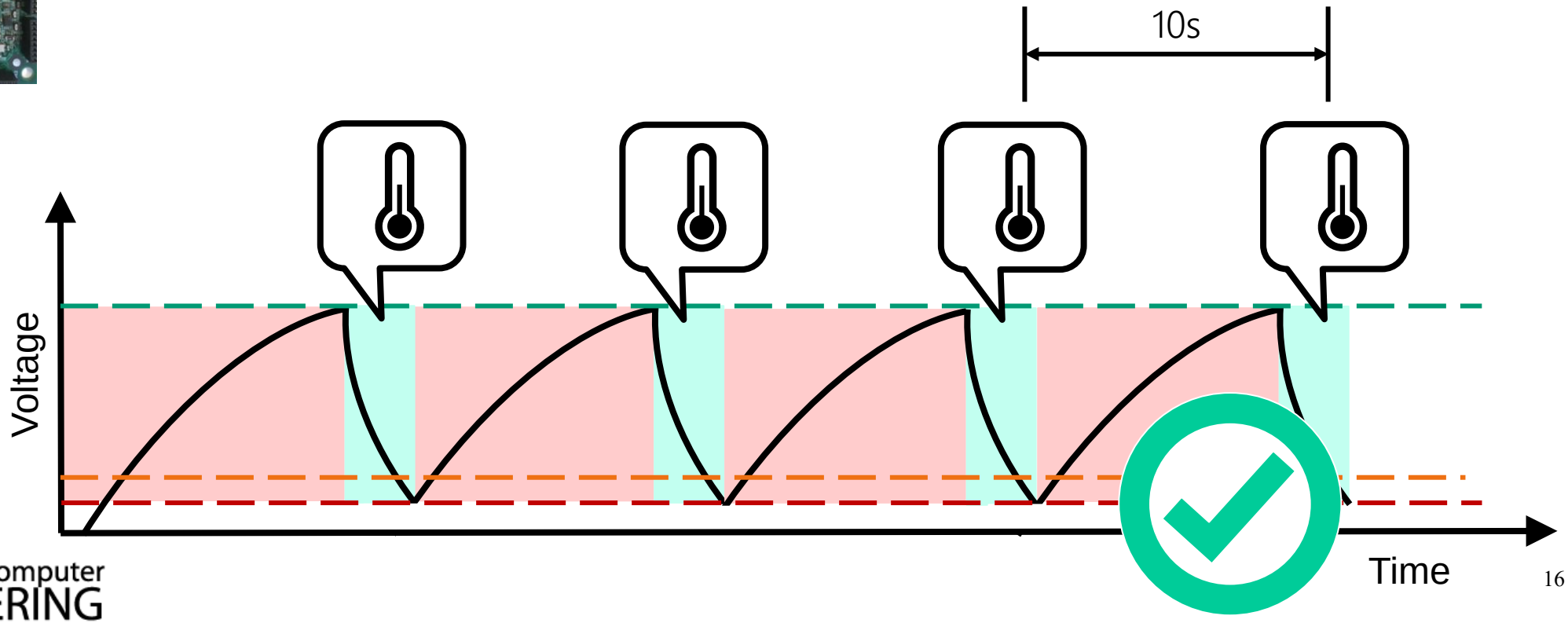
Power Failures Complicate Periodic Execution



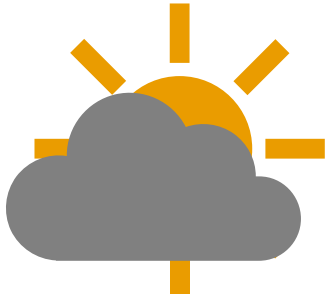
Power Failures Complicate Periodic Execution

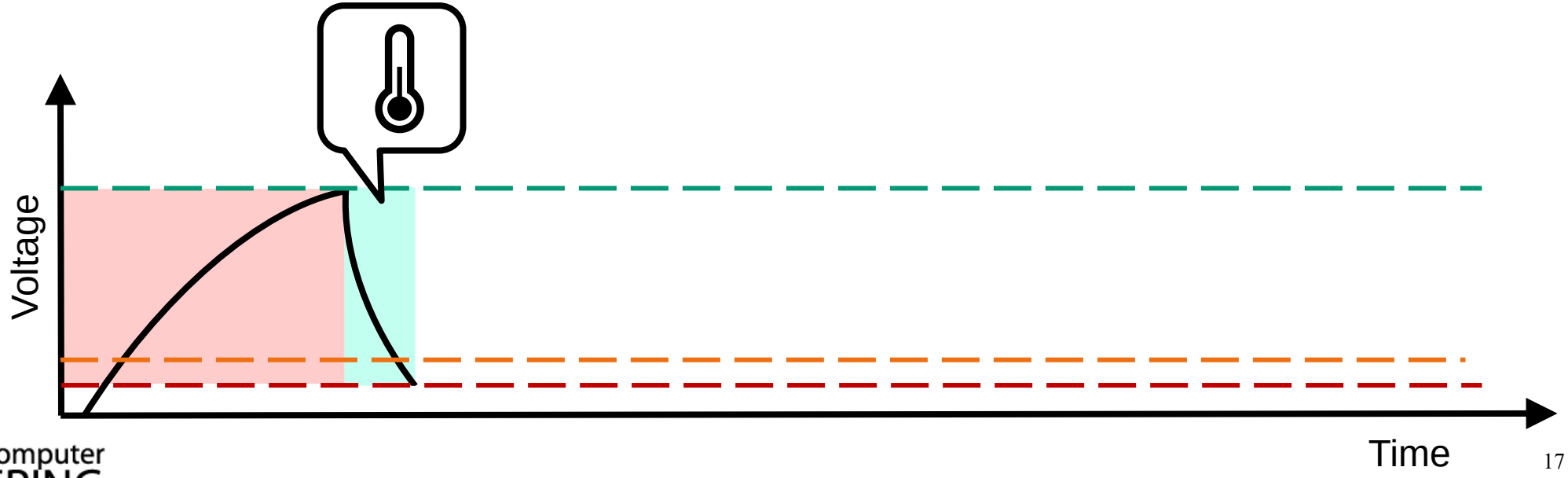


Power Failures Complicate Periodic Execution

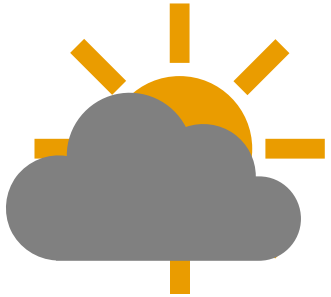


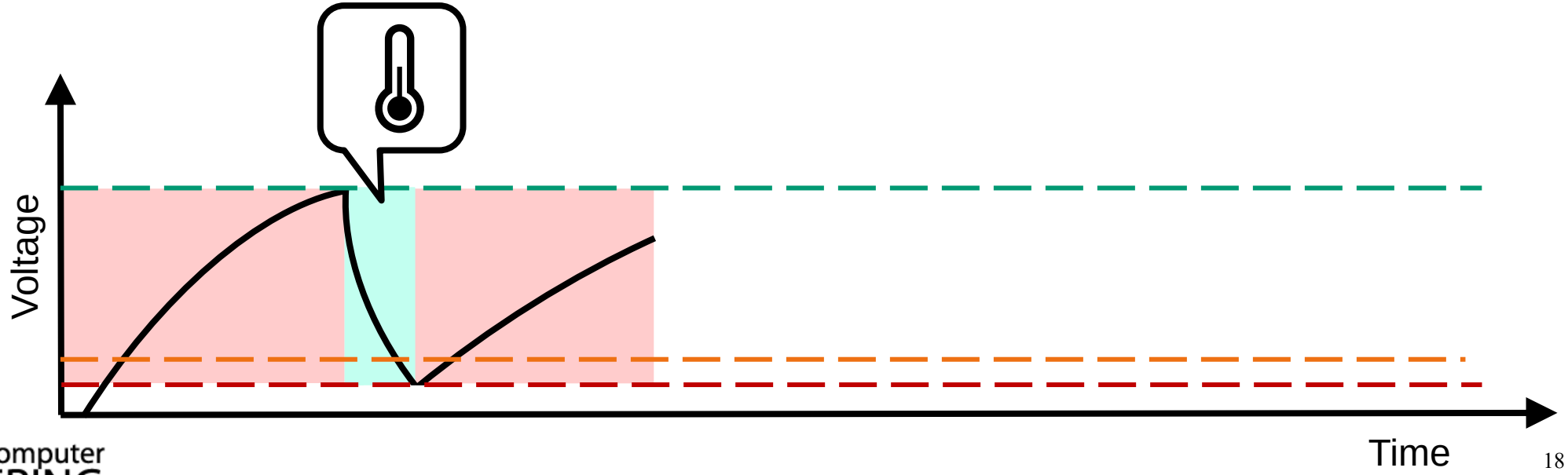
Power Failures Complicate Periodic Execution



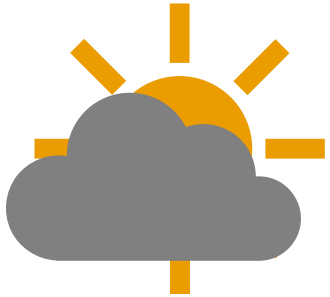


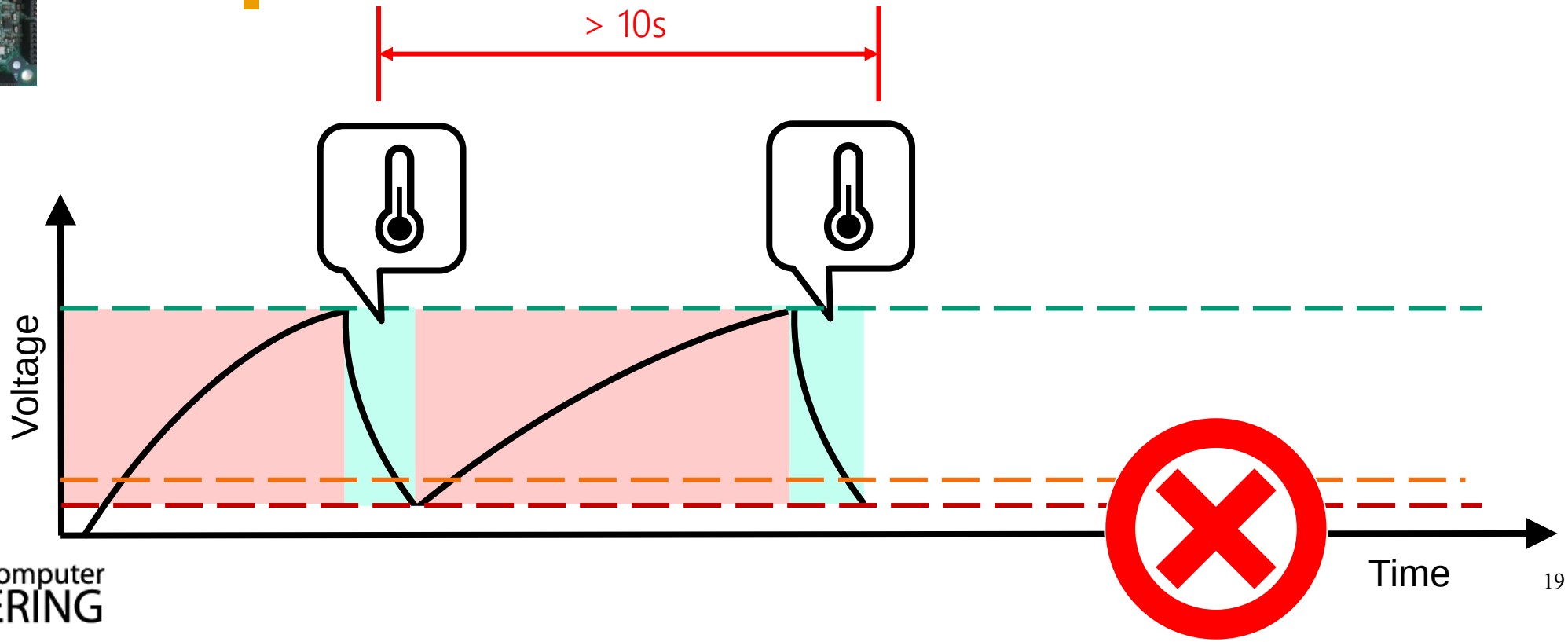
Power Failures Complicate Periodic Execution



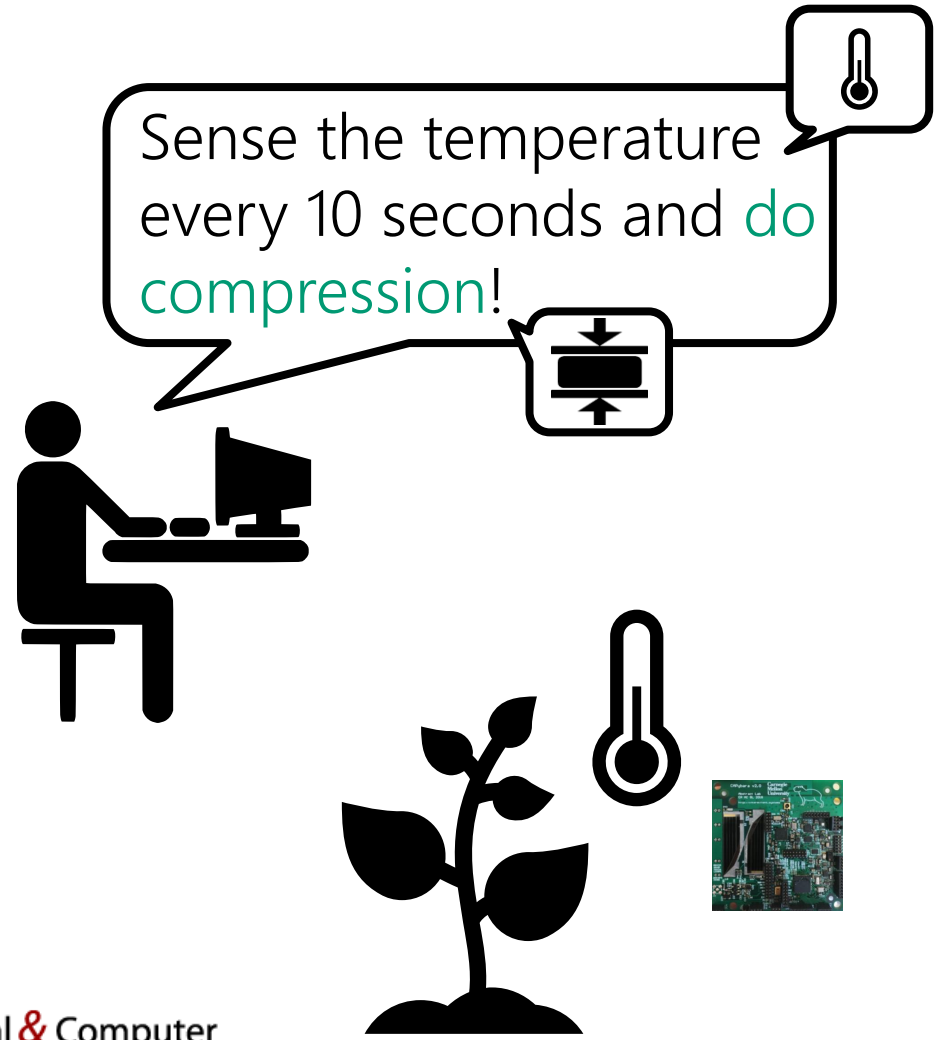


Power Failures Complicate Periodic Execution

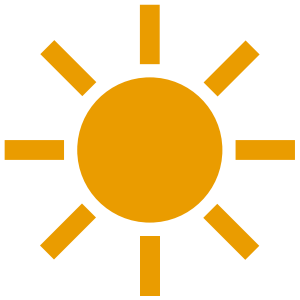


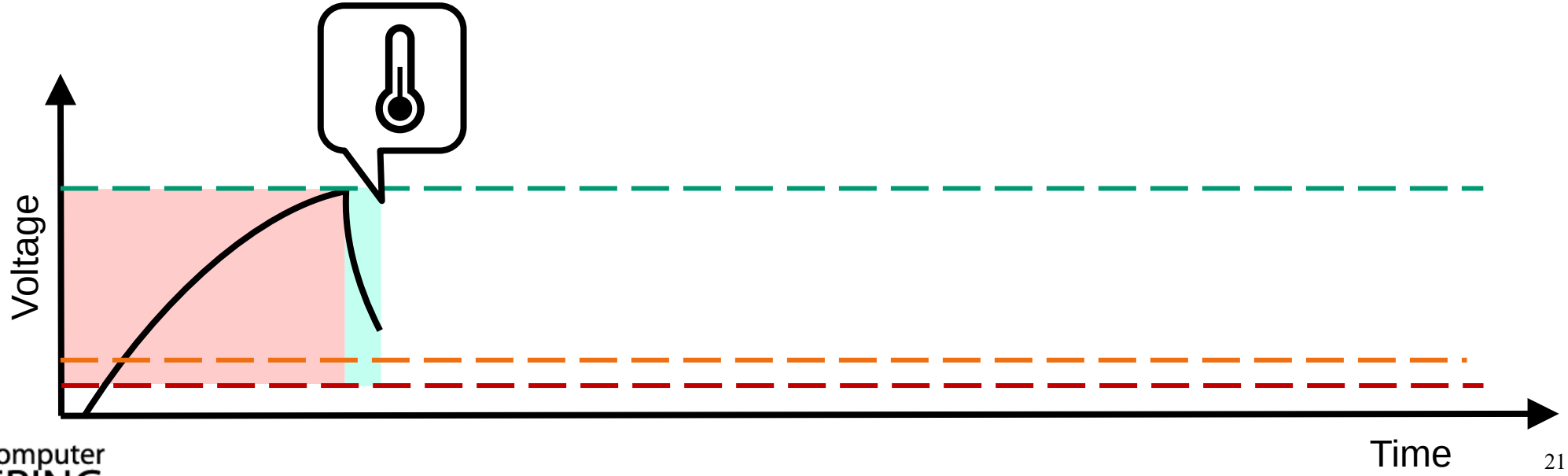


Concurrent Execution Complicate Periodic Execution

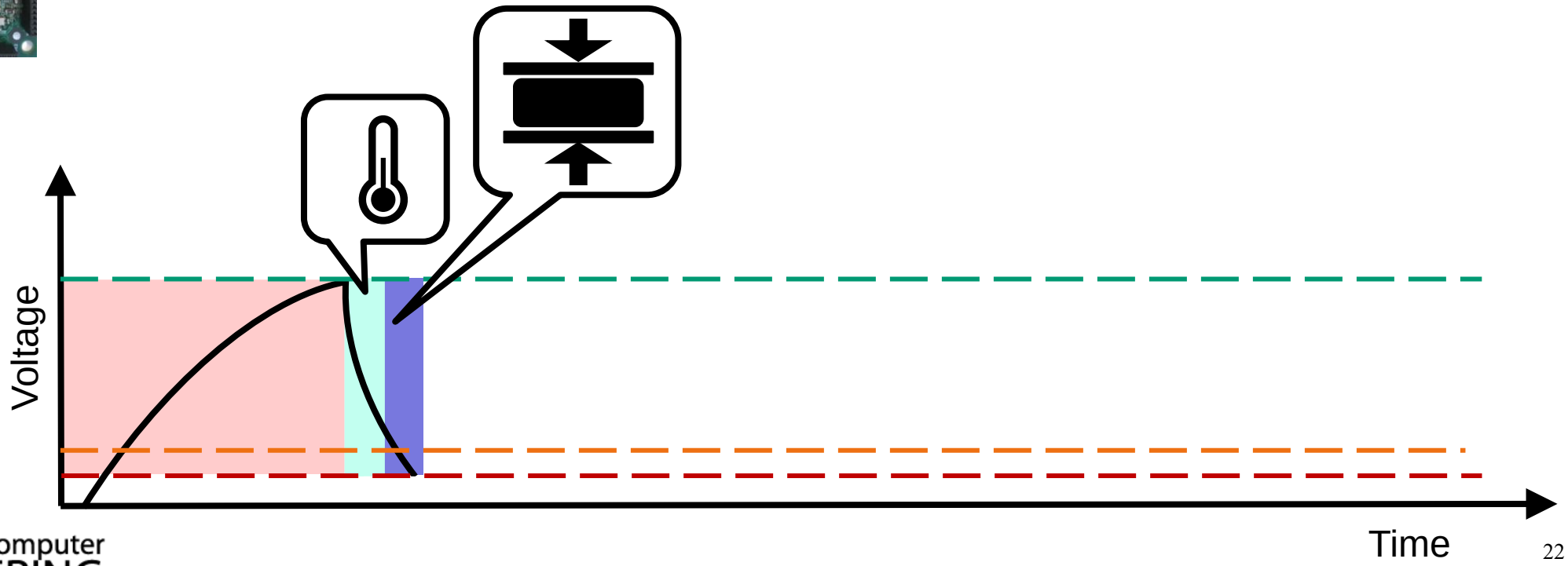


Concurrent Execution Complicate Periodic Execution

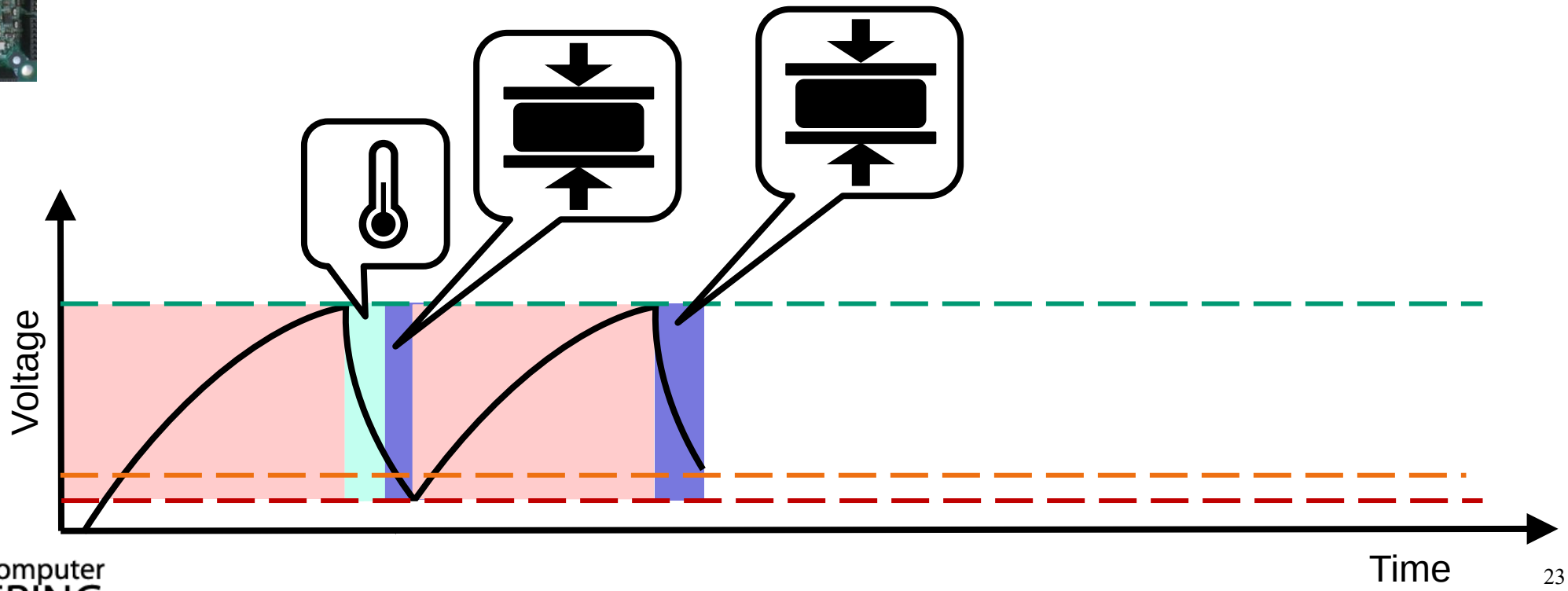




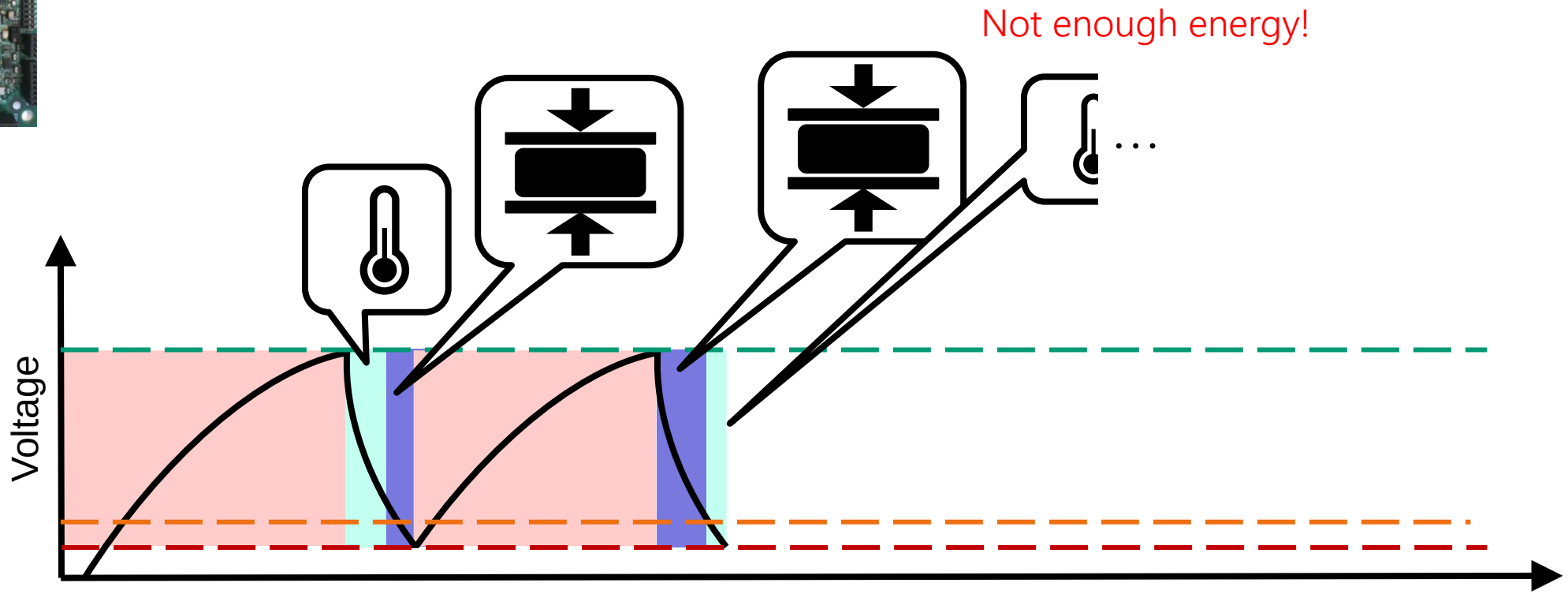
Concurrent Execution Complicate Periodic Execution



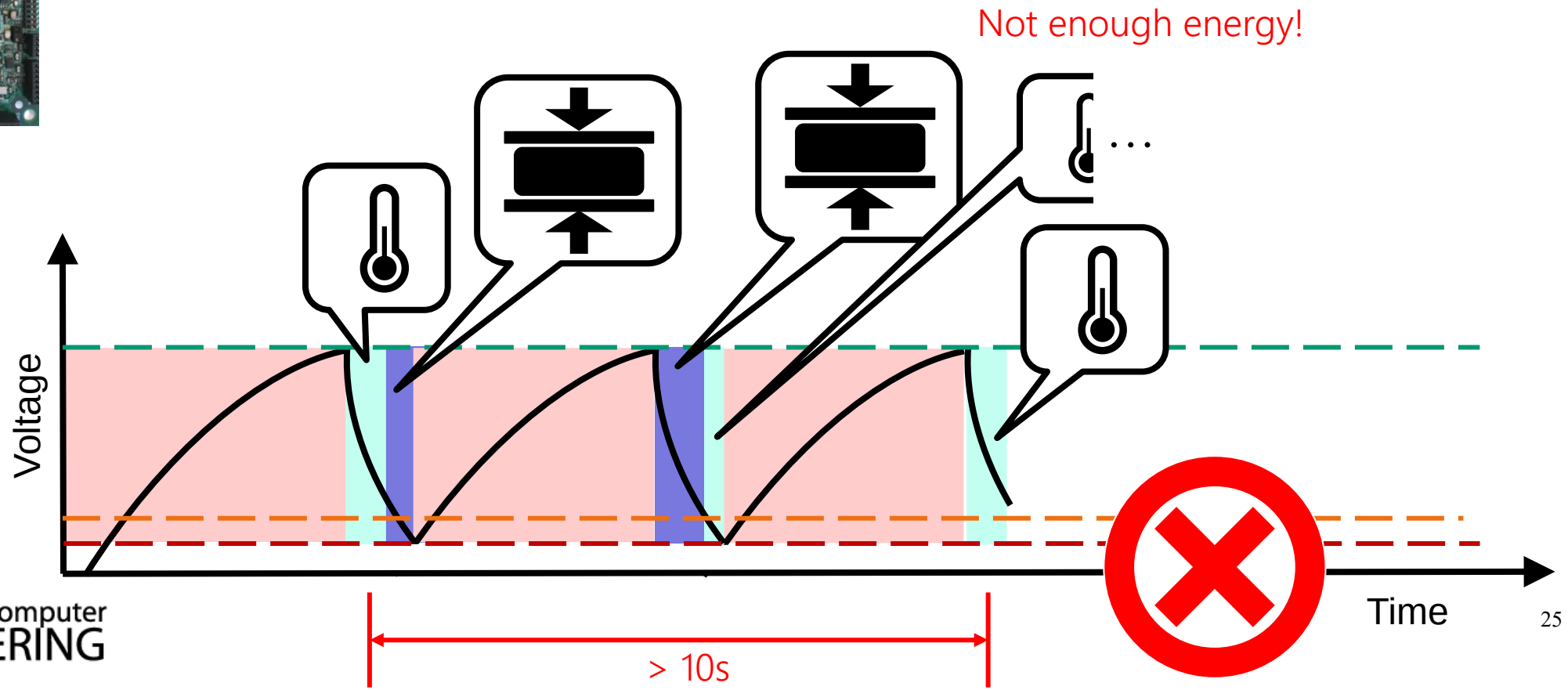
Concurrent Execution Complicate Periodic Execution



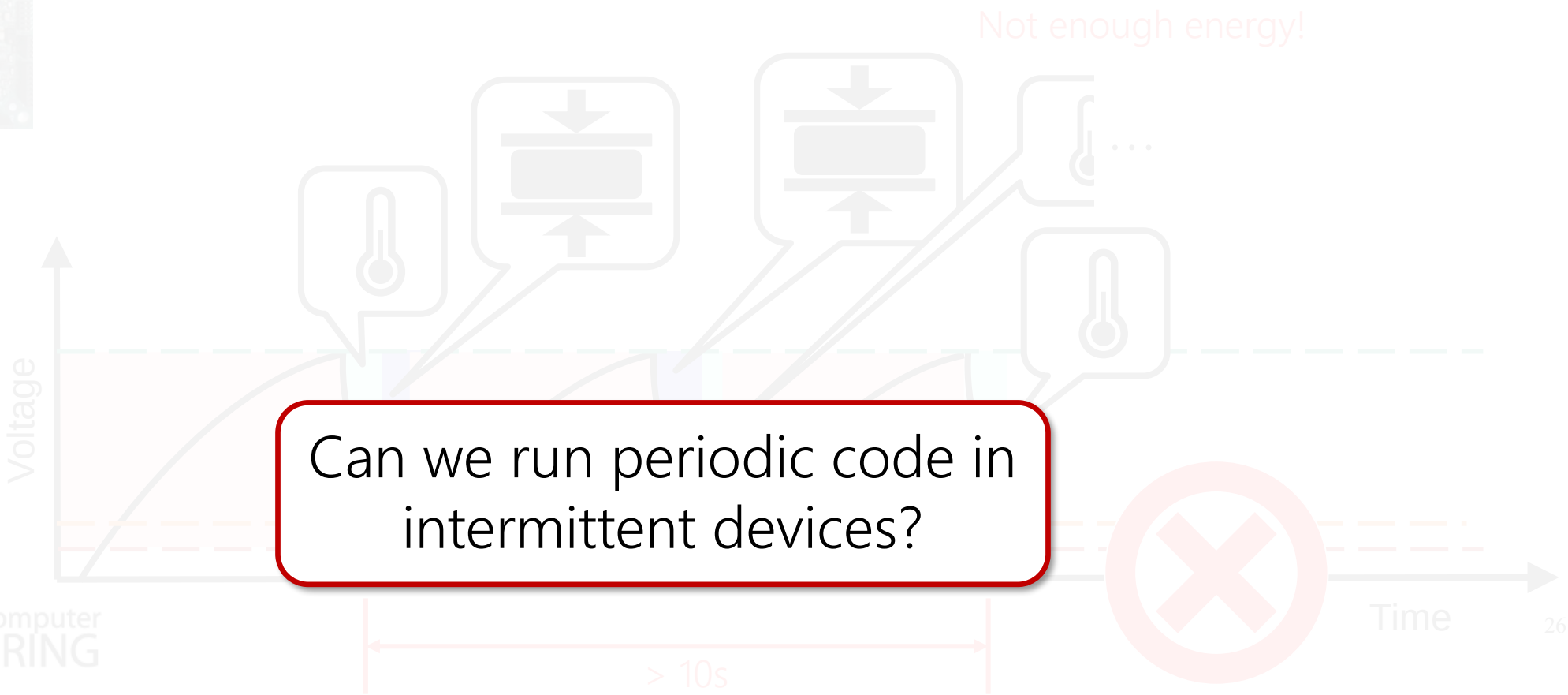
Concurrent Execution Complicate Periodic Execution



Concurrent Execution Complicate Periodic Execution



Concurrent Execution Complicate Periodic Execution



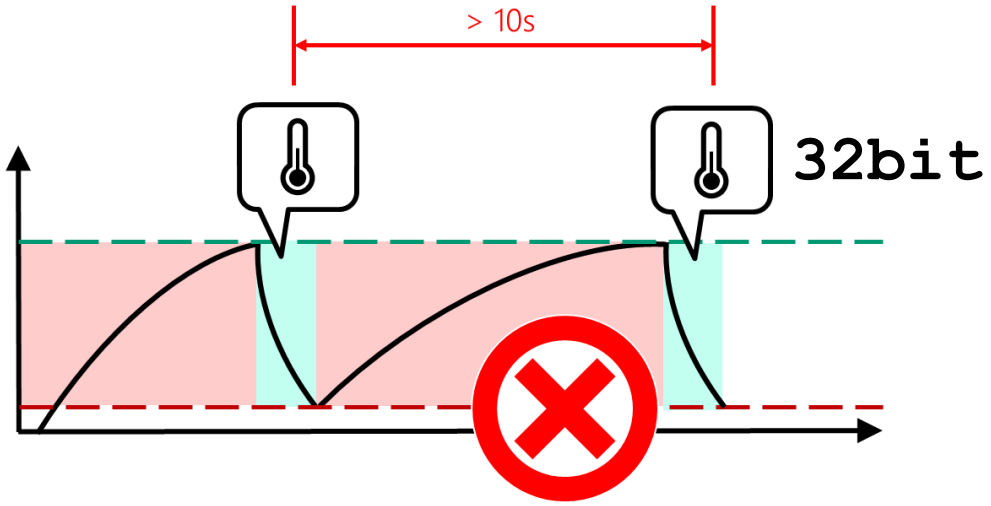
Can we run periodic code in intermittent devices?

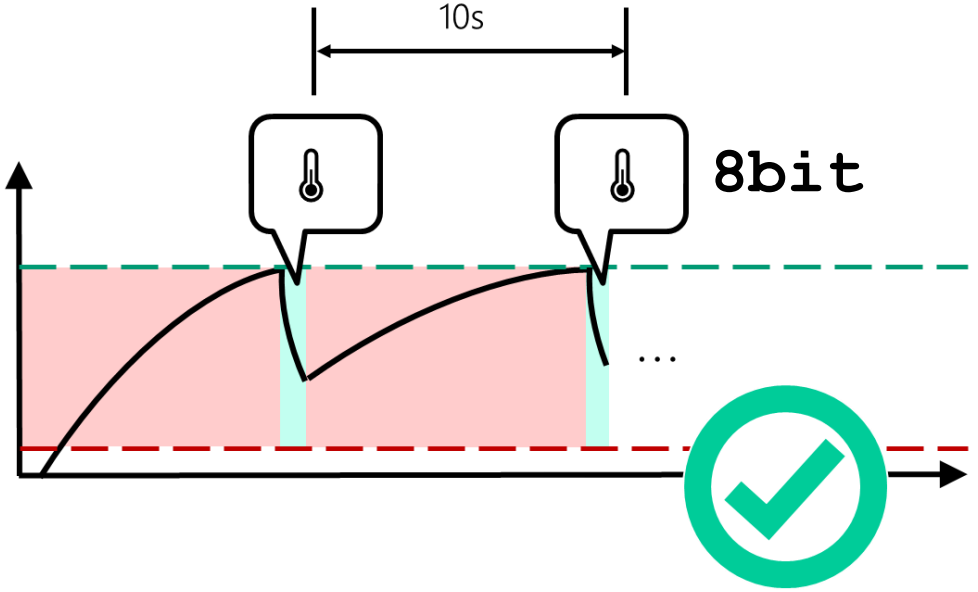
Outline

- Challenge 1. Periodic Execution
- **Solution 1. CatNap**
- Challenge 2. Atomic Execution
- Solution 2. Samoyed
- Future Work

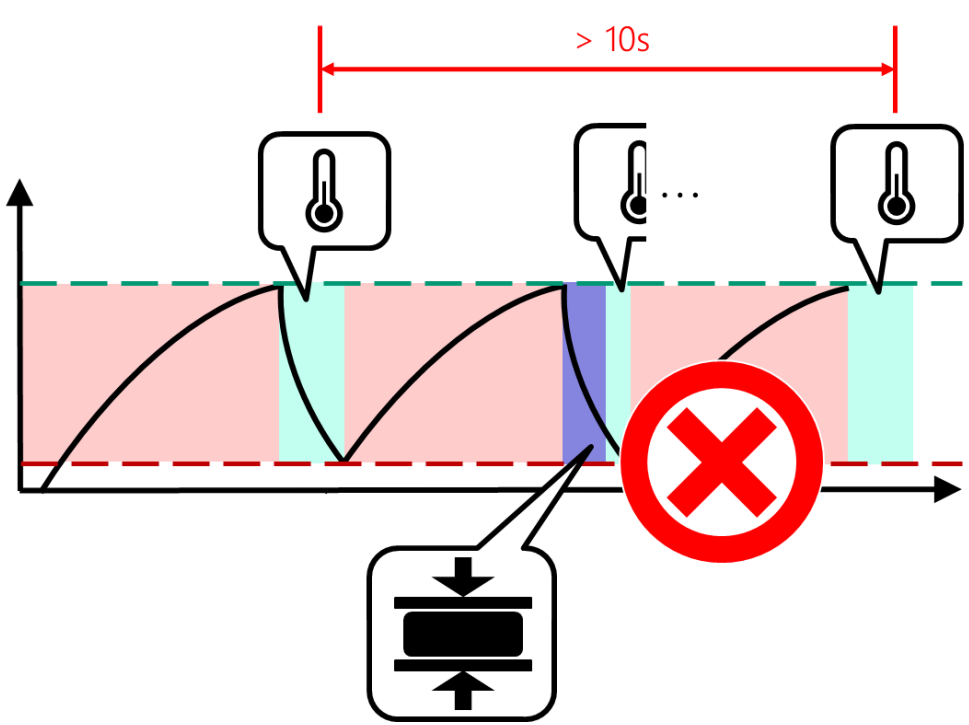
* Kiwan Maeng and Brandon Lucia, Adaptive Low-Overhead Scheduling for Periodic and Reactive Intermittent Execution. PLDI 2020

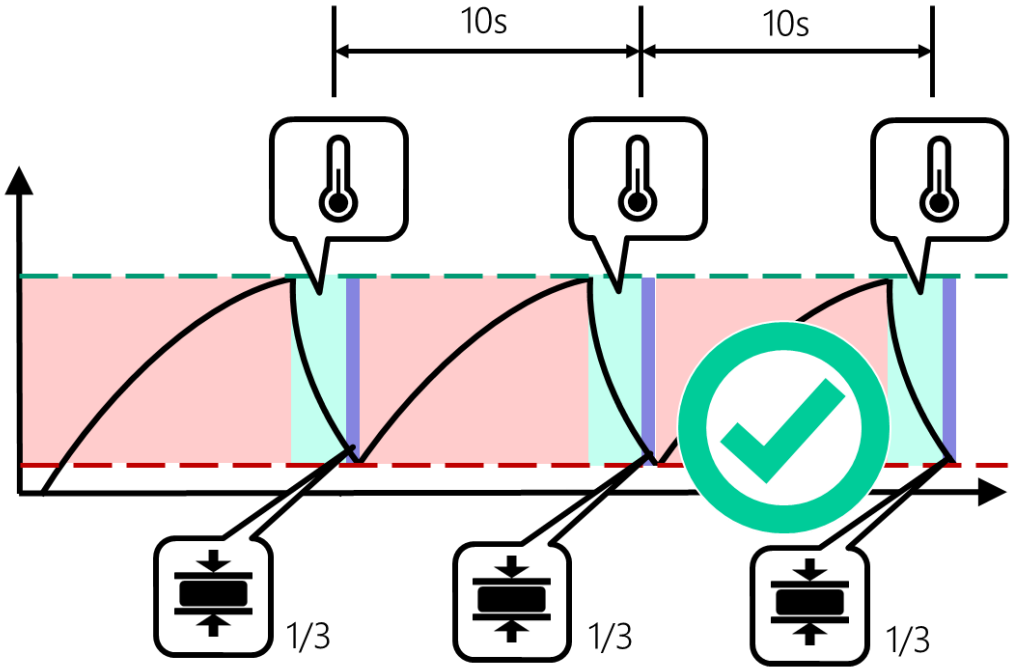
Idea 1: Quality degradation





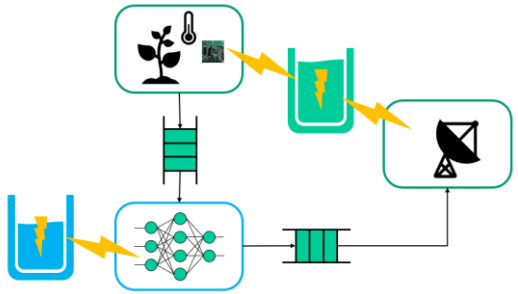
Idea 2: Schedule Code Execution and **Recharging**



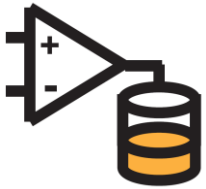


CatNap: Computation and **Recharge** Scheduler

Programming Model

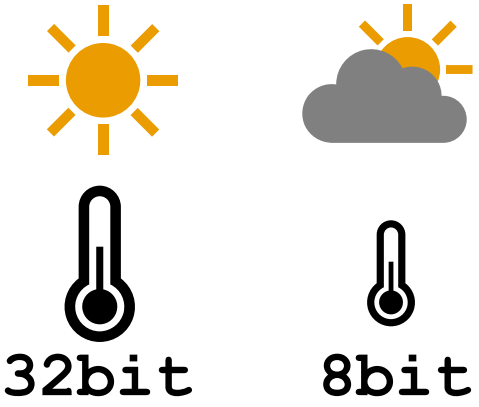


Scheduler



$$\sum_{i=1}^{n} \frac{c_i}{t_i} \leq 1$$

Feasibility Test

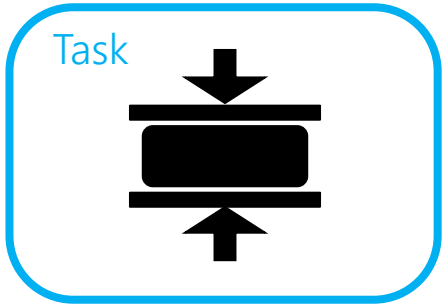


Degradation Logic

Programmer Specifies Time-critical *Events* and Time-insensitive *Tasks*

Event

- Code with a time requirement (periodic/reactive requirement)
- Short, atomic
- e.g., sensor read, communication, …

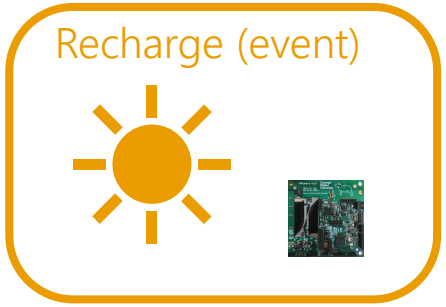


Task

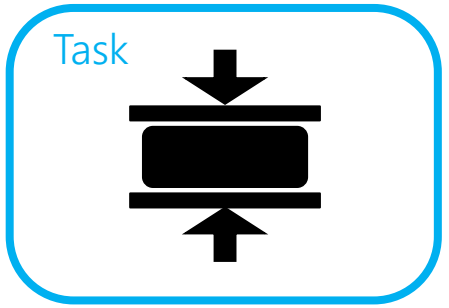
- Code without a time requirement
- Long, interruptible
- e.g., compute

CatNap Schedules According to Priorities

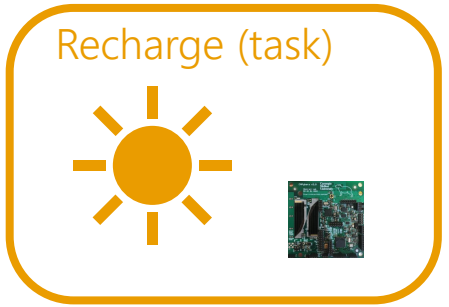
Events
High priority



Event buffer recharge
Middle priority

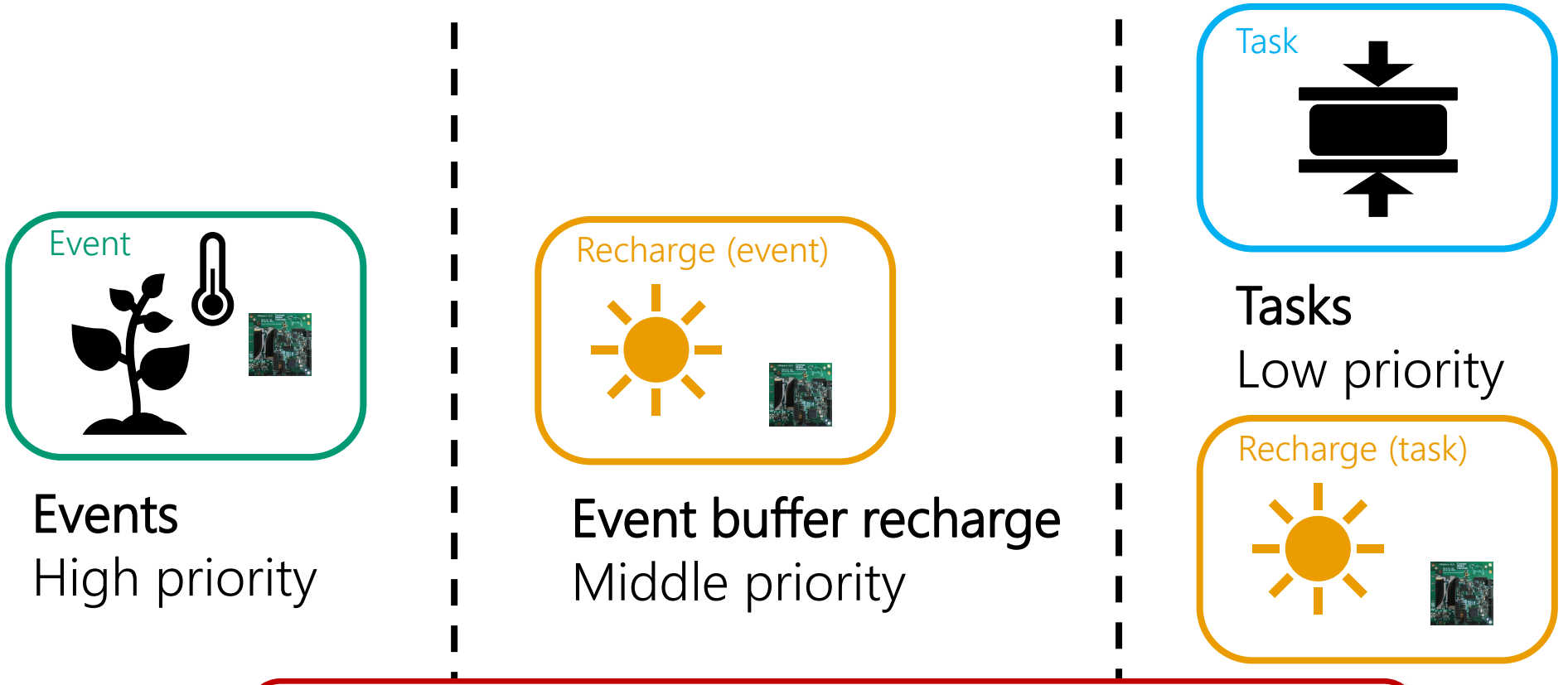


Tasks
Low priority



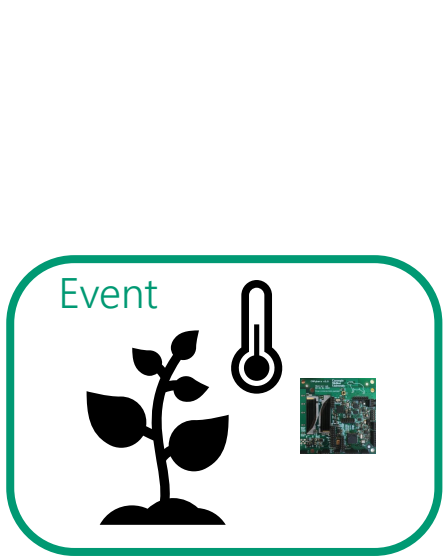
Task buffer recharge
Low priority

CatNap Schedules According to Priorities

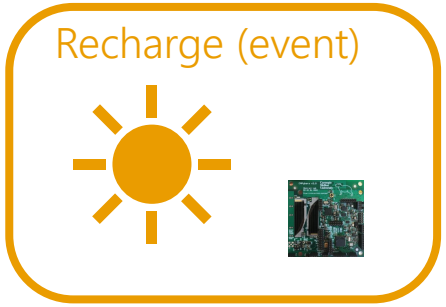


If a set of events and recharges are schedulable, CatNap can always schedule them! (proof: paper)

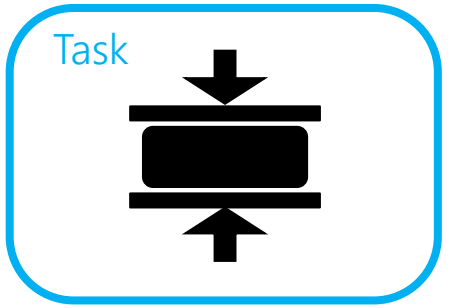
CatNap Schedules According to Priorities



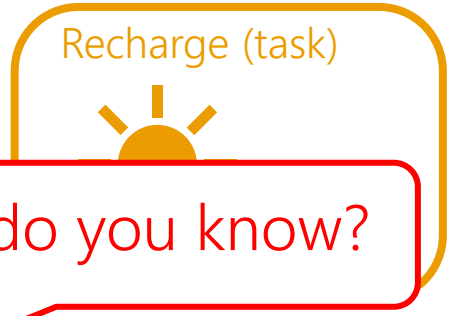
Events
High priority



Event buffer recharge
Middle priority



Tasks
Low priority



fer recharge
ority

How do you know?

If a set of events and recharges are schedulable,
CatNap can always schedule them! (proof: paper)

Electrical & Computer ENGINEERING

A *Feasibility Test* Checks If Events are Schedulable

DETAIL

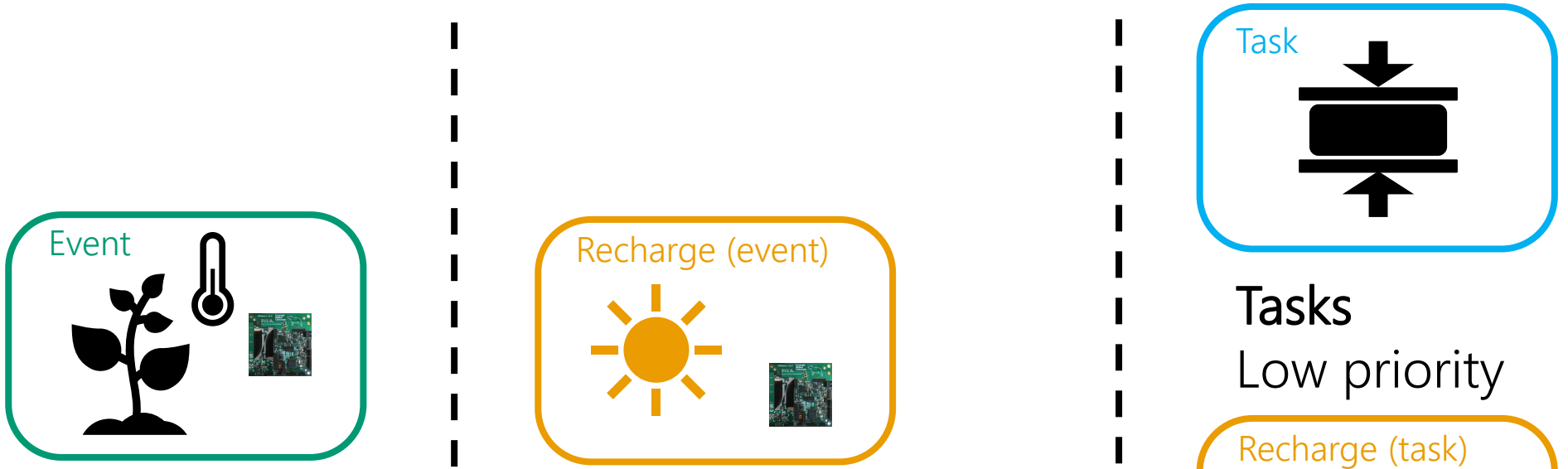
Theorem

Assume an event $\varepsilon_i \in \{\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_n\}$ uses energy e_i and has a period of t_i. If an incoming power is R and $c_i = \frac{e_i}{R}$, the events and charges can be scheduled if:

$$\sum_{i=1}^{n} \frac{c_i}{t_i} \leq 1$$

Full proof is provided as an Appendix of the paper.

CatNap Schedules According to Priorities



Events
High priority

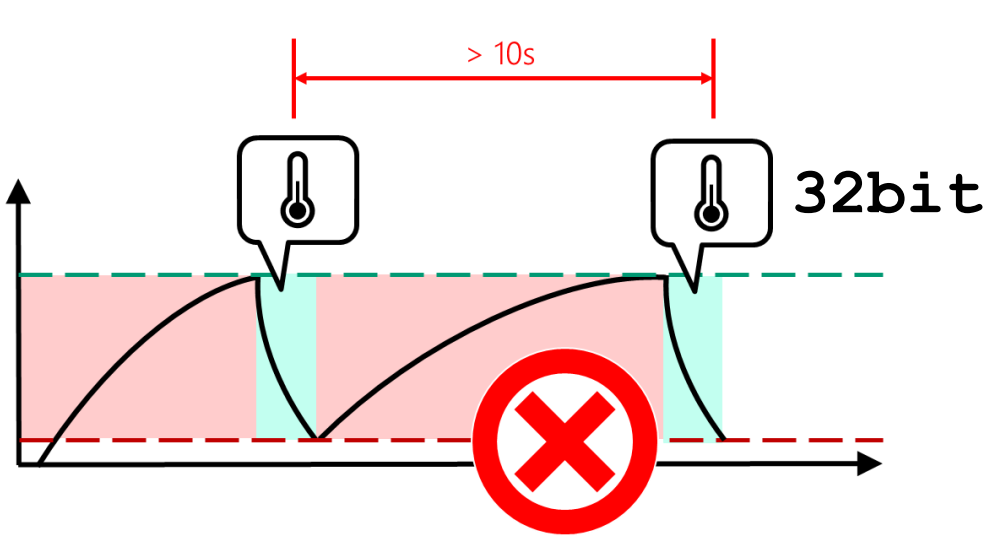
Event buffer recharge
Middle priority

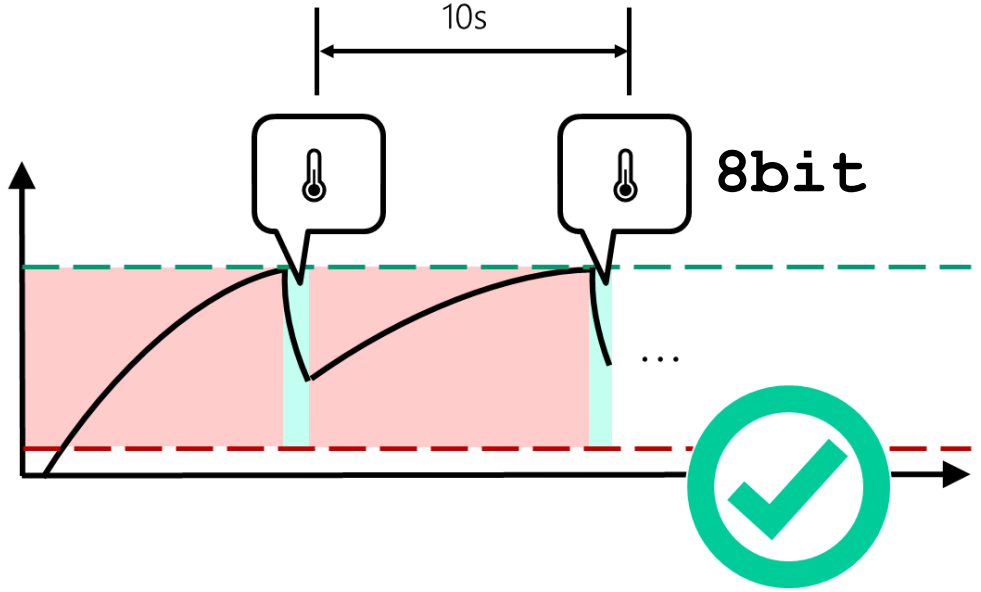
Tasks
Low priority

What if it's not?

If a set of events and recharges are schedulable,
CatNap can always schedule them! (proof: paper)

Quality Degradation Makes Infeasible Schedule Feasible





Programmer provides set of degradation rules

Evaluation: CatNap Enables Periodic Execution

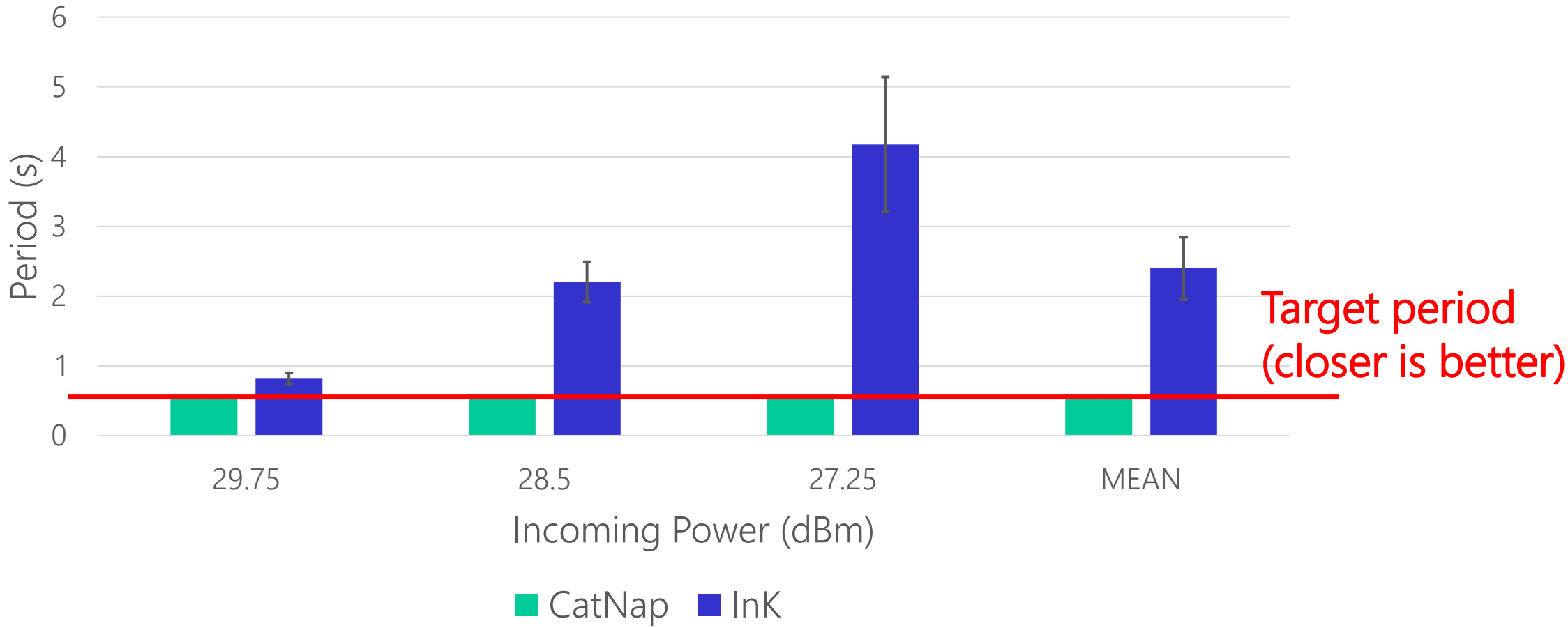
- Ran periodic and compute-intensive workloads concurrently

Periodic temperature sensing every 0.57s (event)

Downsampling with a square filter (task)

- Ran by harvesting RF energy
- Compared with InK [Yıldırım 2018]

Evaluation: CatNap Enables Periodic Execution

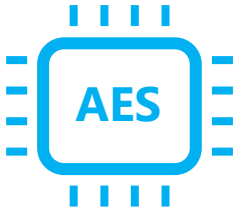


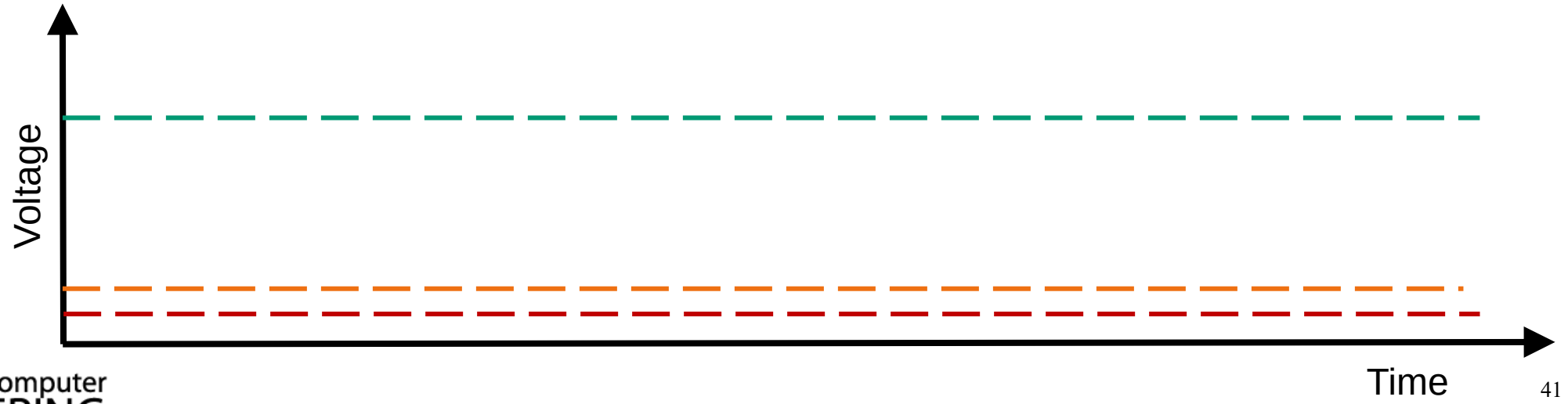
Outline

- Challenge 1. Periodic Execution
- Solution 1. CatNap
- **Challenge 2. Atomic Execution**
- Solution 2. Samoyed
- Future Work

Power Failure Breaks Atomicity

Workload

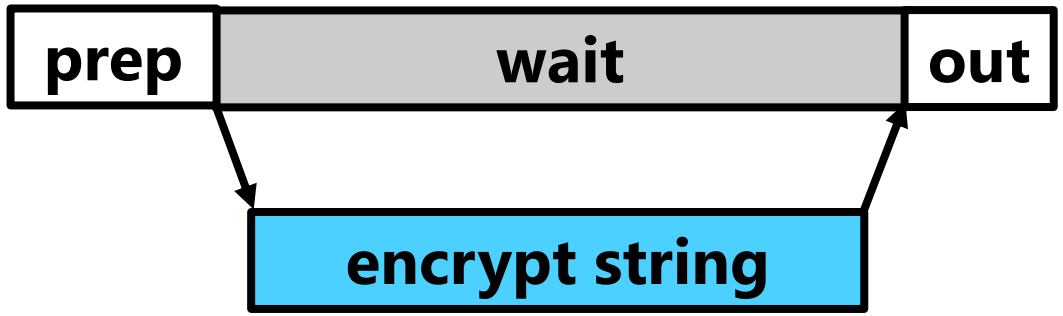


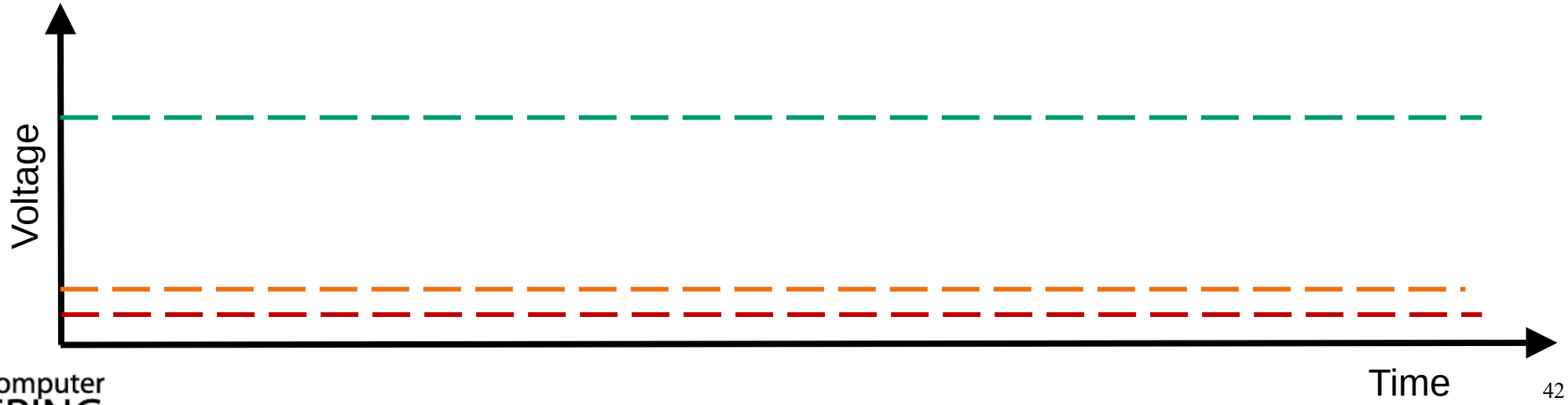


Power Failure Breaks Atomicity

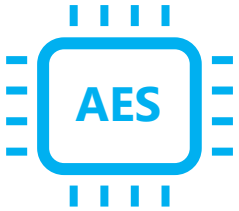
AES

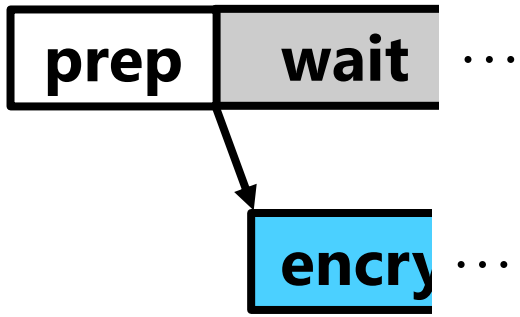
Workload



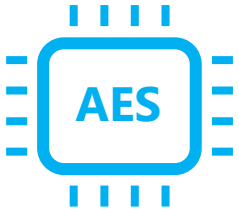


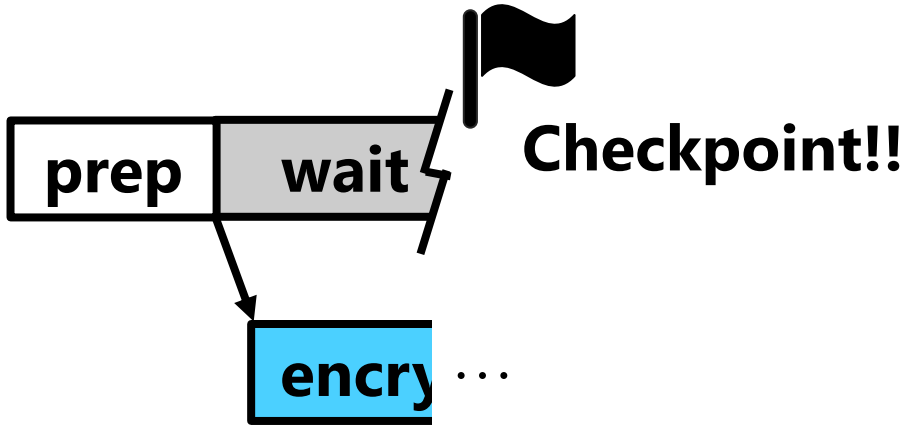
Power Failure Breaks Atomicity

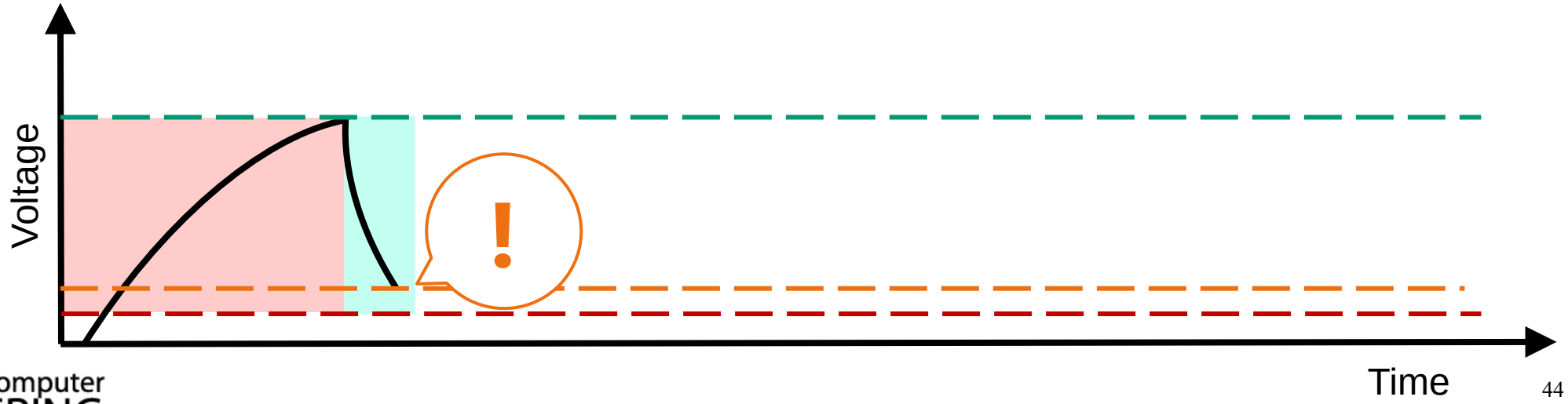




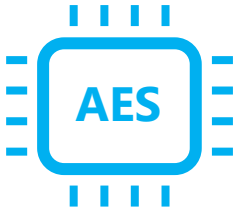
Power Failure Breaks Atomicity

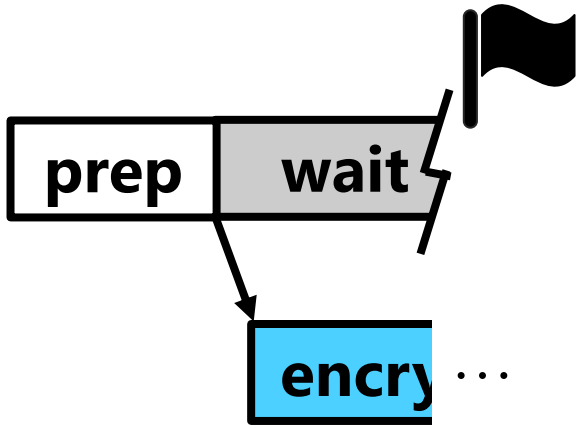






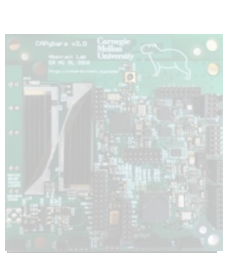
Power Failure Breaks Atomicity

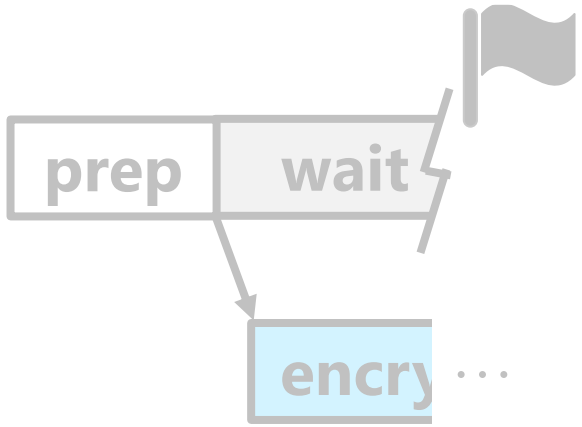




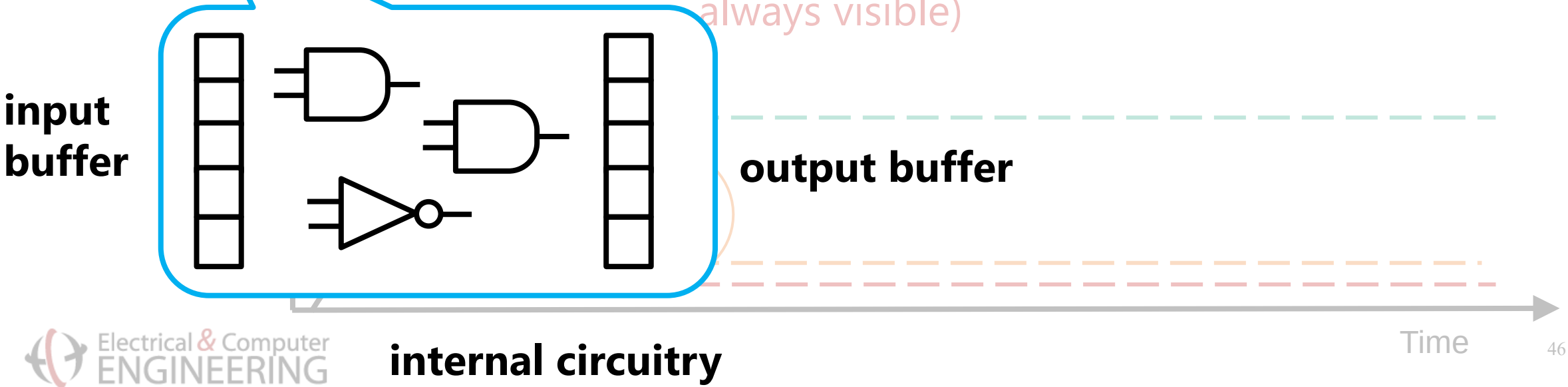
Cannot checkpoint…
(peripheral state not always visible)

Power Failure Breaks Atomicity

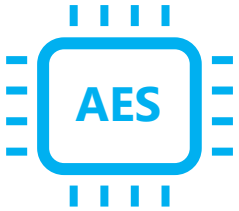


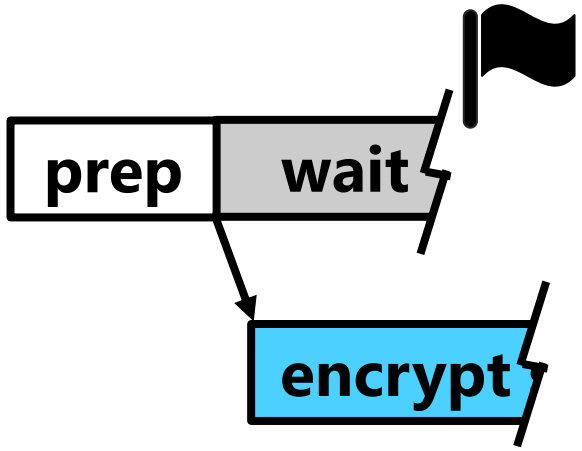


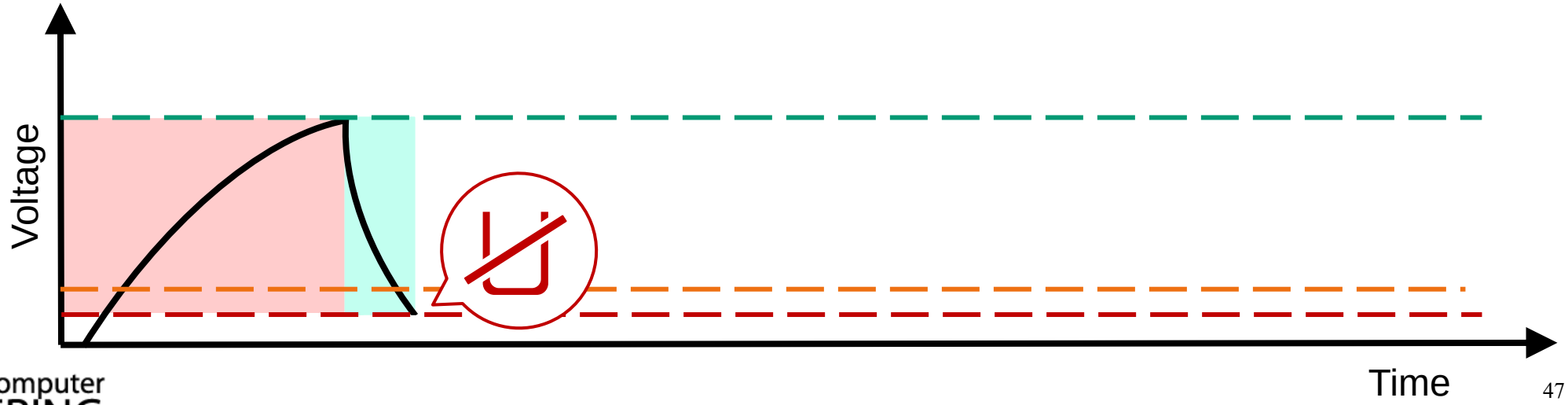
Cannot checkpoint… (peripheral state not always visible)



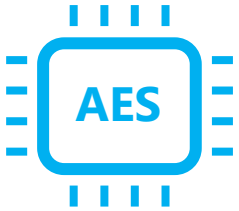
Power Failure Breaks Atomicity

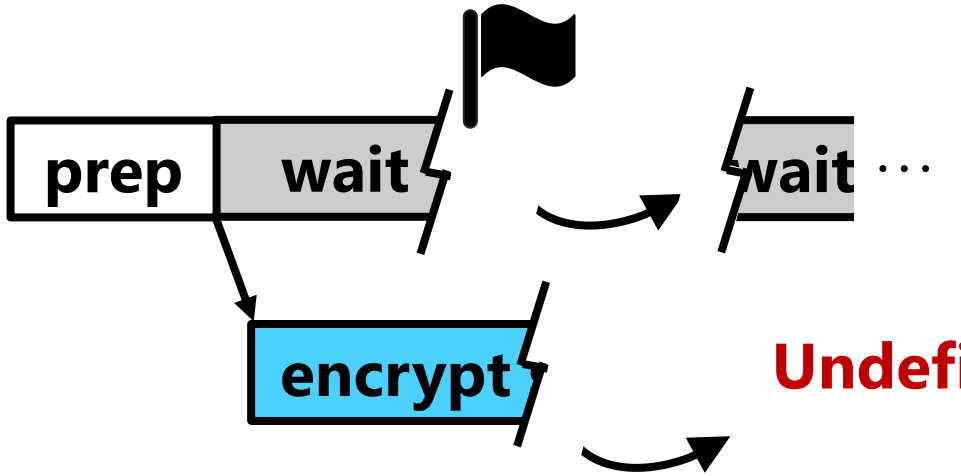




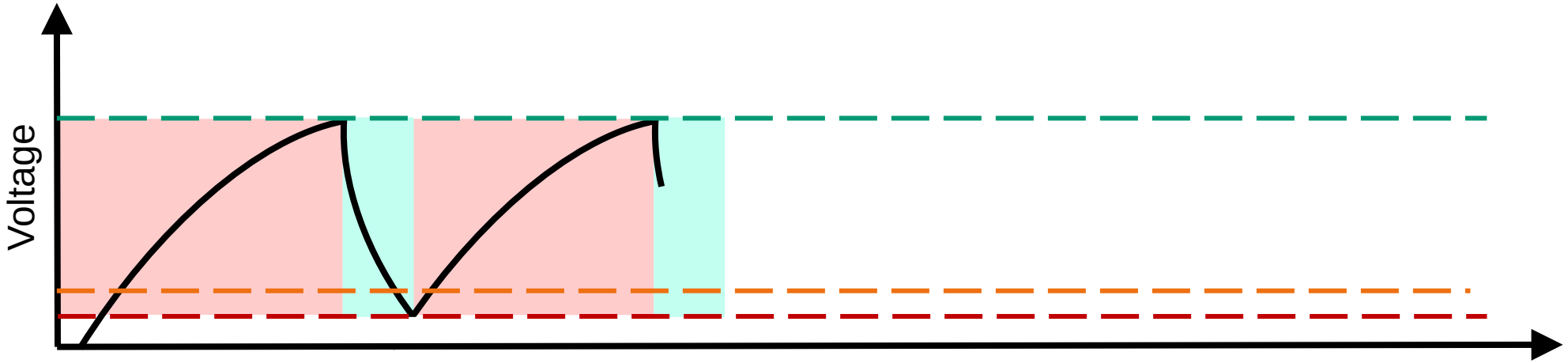


Power Failure Breaks Atomicity





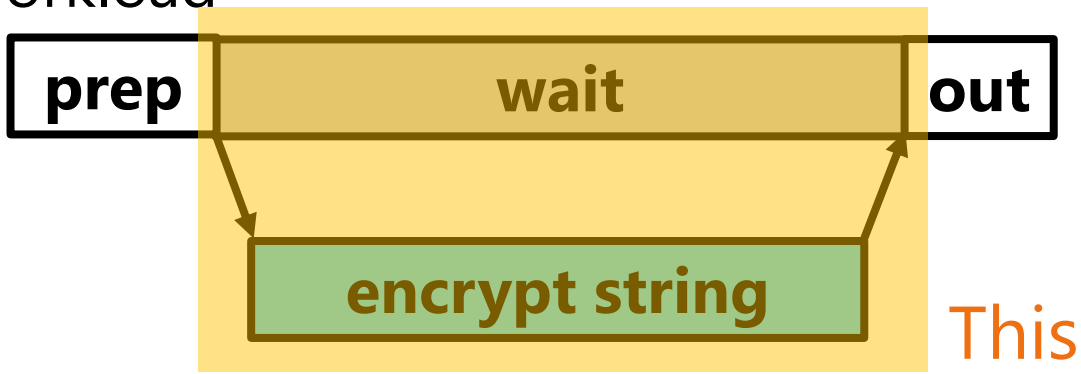
Undefined state!



Power Failure Breaks Atomicity

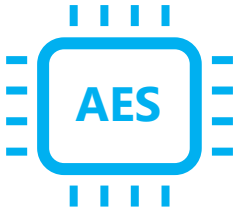
AES

Workload

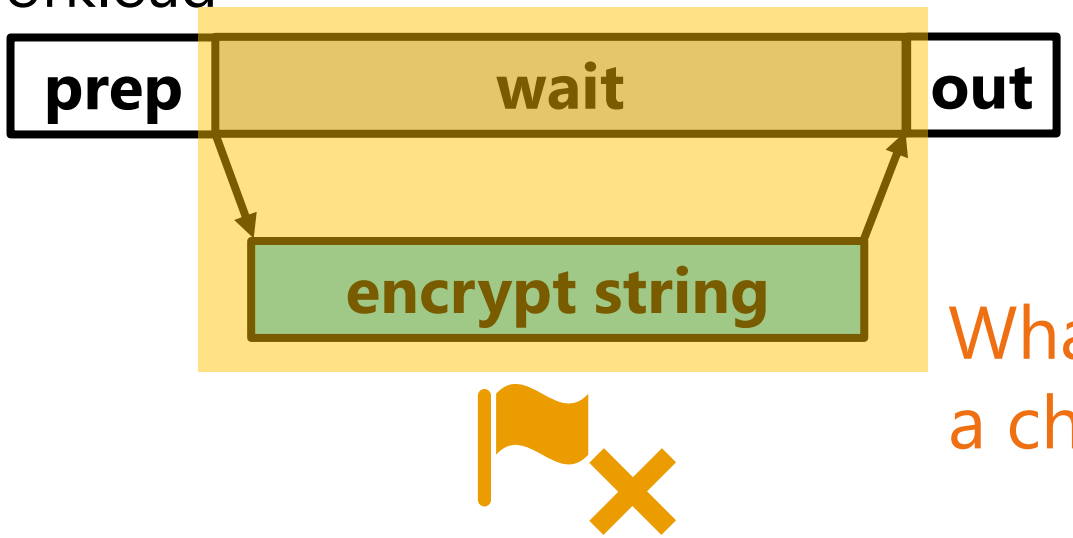


This region must be **atomic**!
(not interrupted by checkpoints/failures)

Initial Attempt: Selectively Disabling Checkpoints

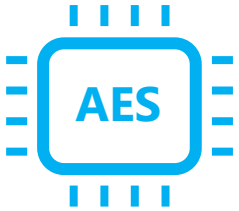


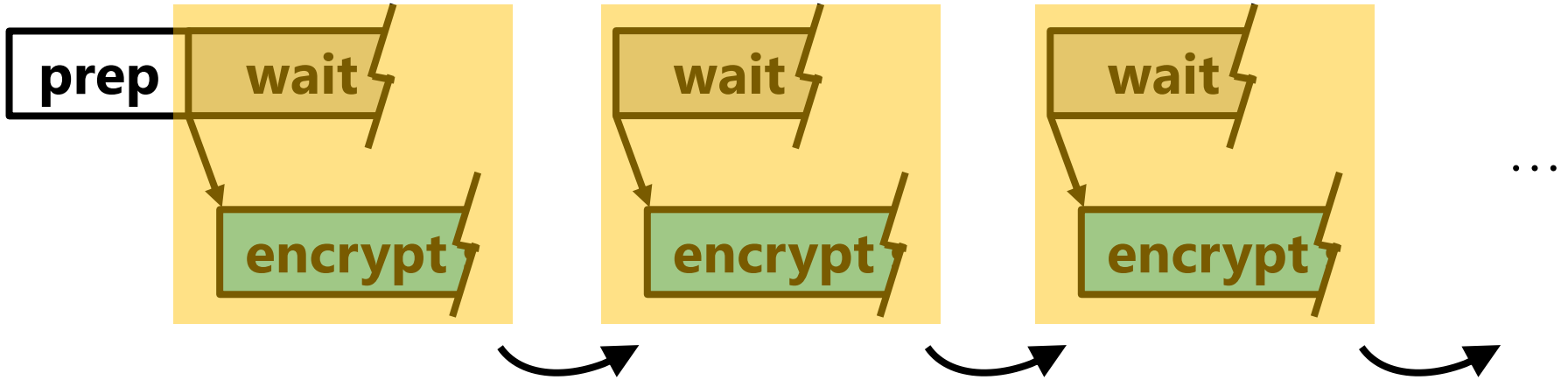
Workload



What if we never collect a checkpoint here?

Initial Attempt: Selectively Disabling Checkpoints





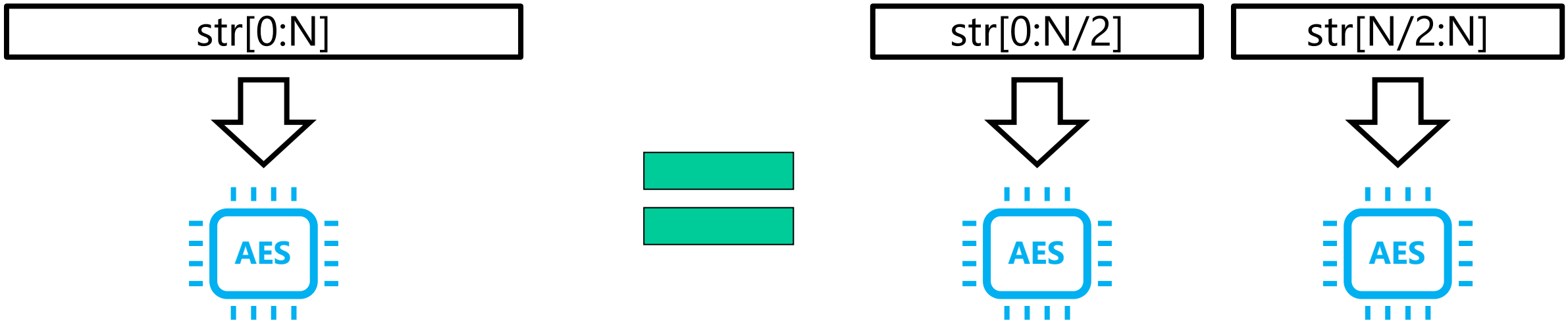
The program never ends!

Outline

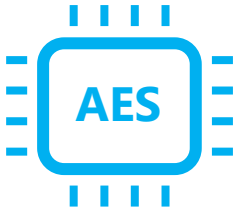
- Challenge 1. Periodic Execution
- Solution 1. CatNap
- Challenge 2. Atomic Execution
- **Solution 2. Samoyed**
- Future Work

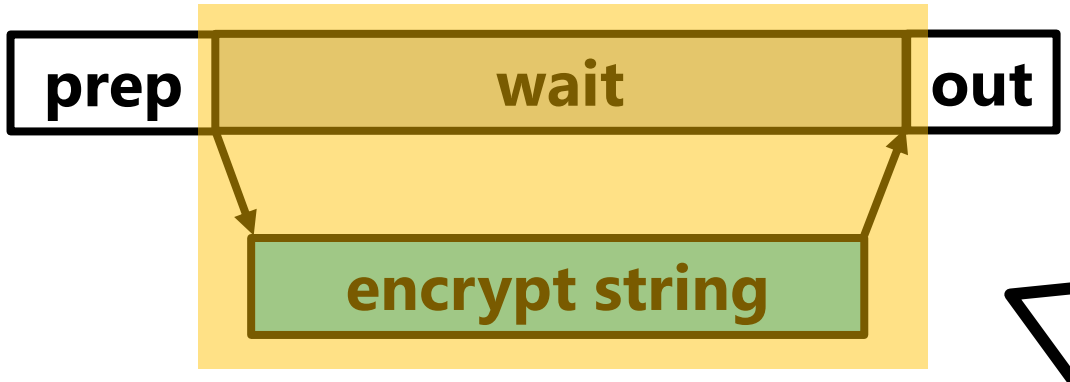
* Kiwan Maeng and Brandon Lucia, **Supporting Peripherals in Intermittent Systems with Just-In-Time Checkpoints**. PLDI 2019

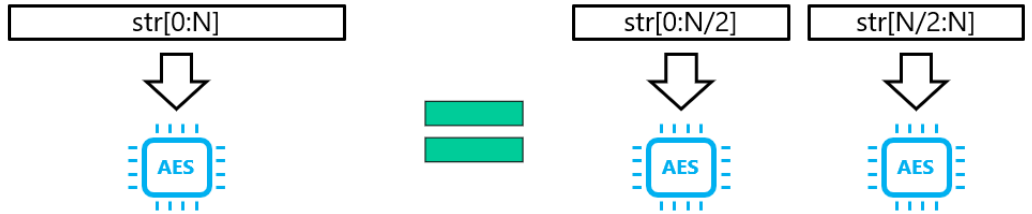
Idea: Peripheral Operations Usually Can Be Decomposed into Smaller Operations

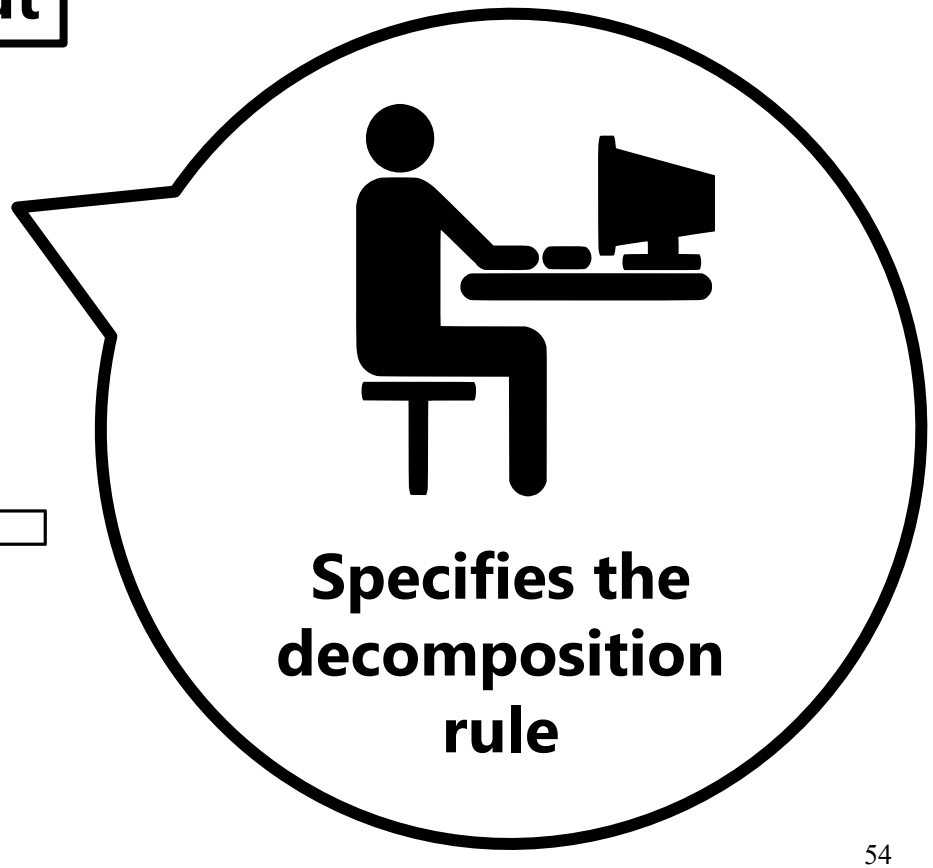


Solution: Peripheral Access Decomposition

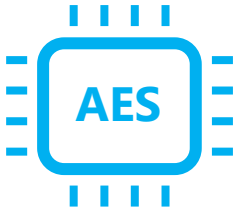


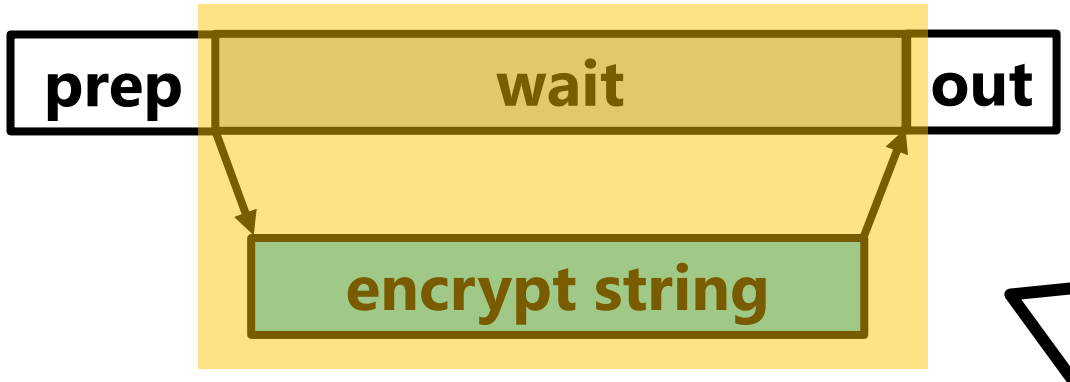


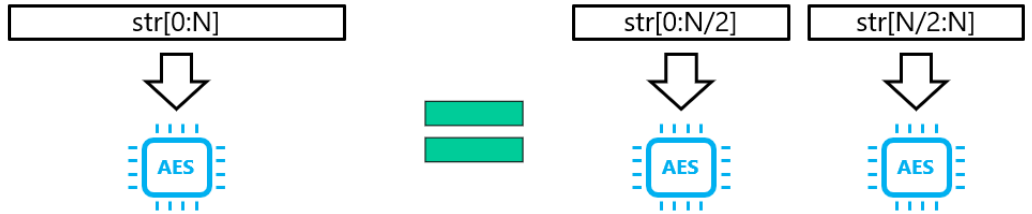


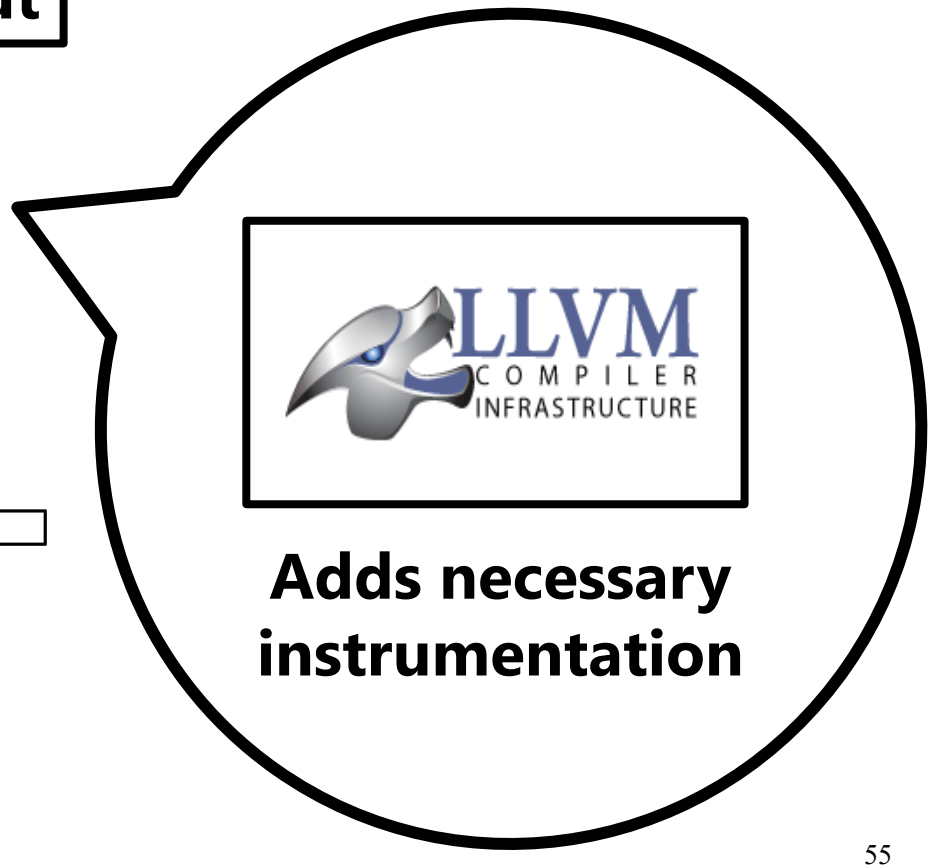


Solution: Peripheral Access Decomposition

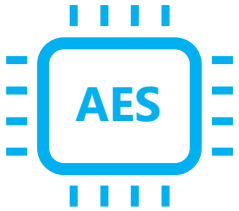


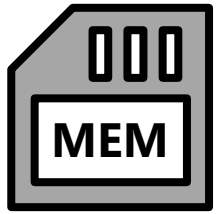


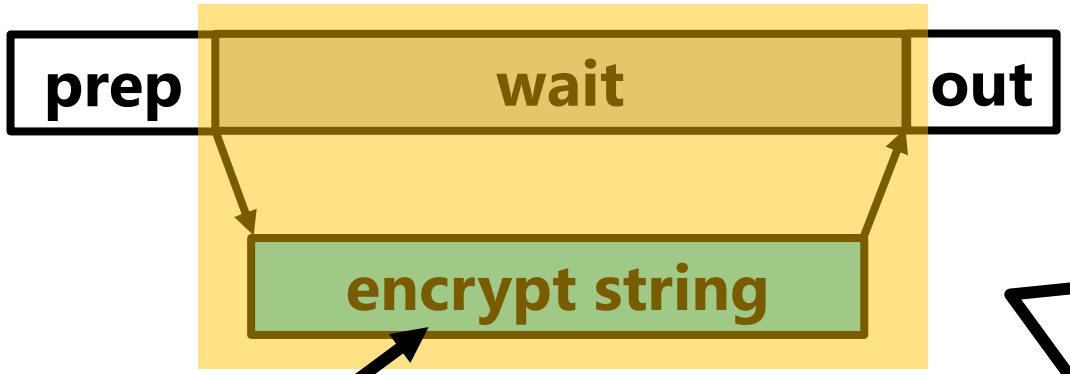




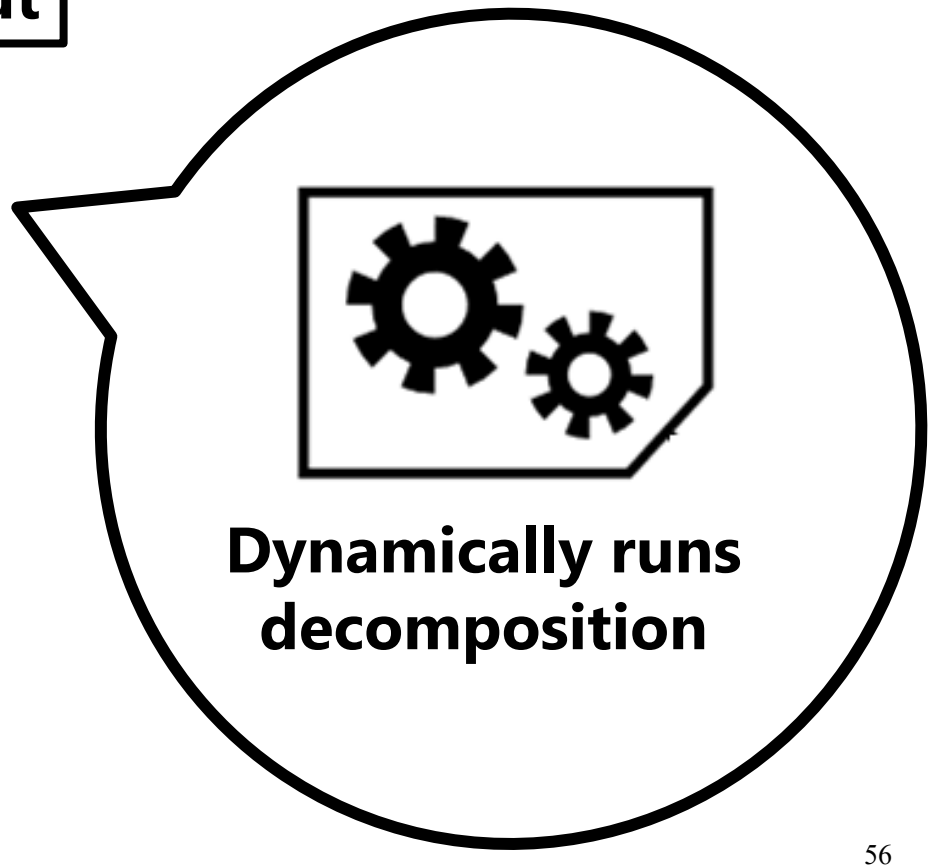
Solution: Peripheral Access Decomposition



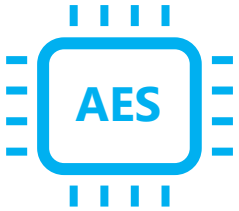


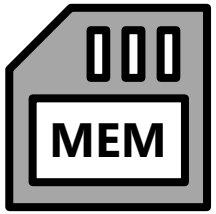


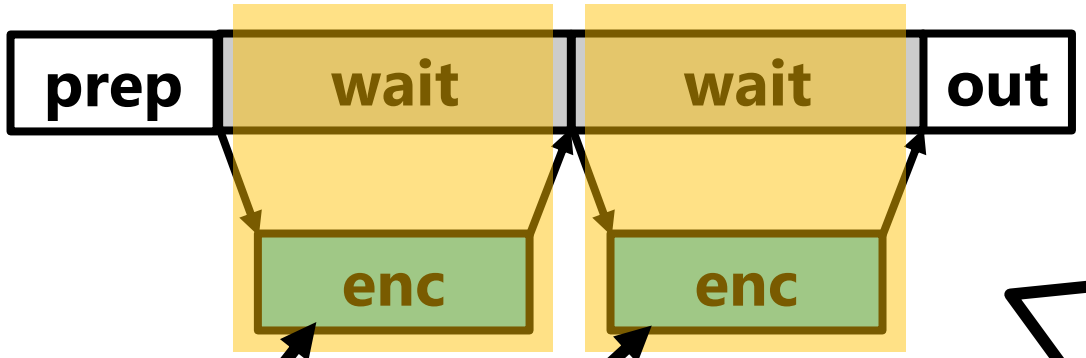

```
Hello, world!
```



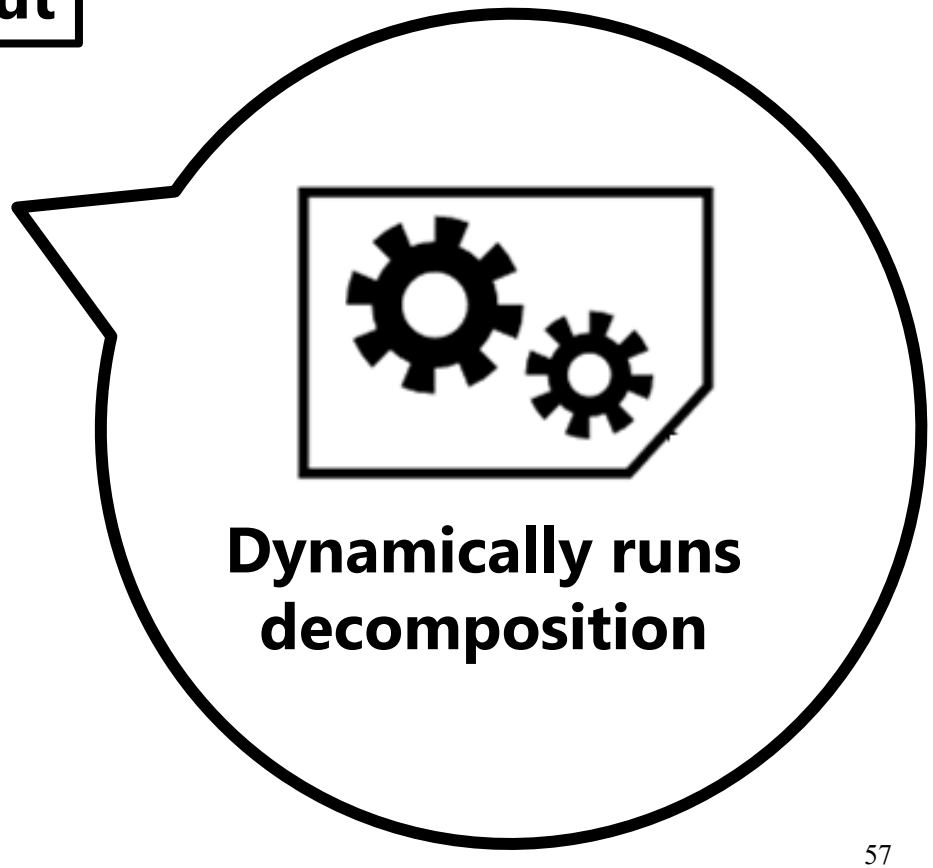
Solution: Peripheral Access Decomposition



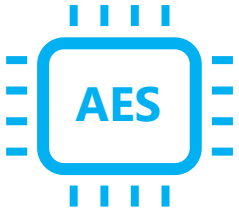


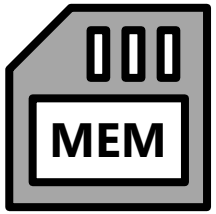


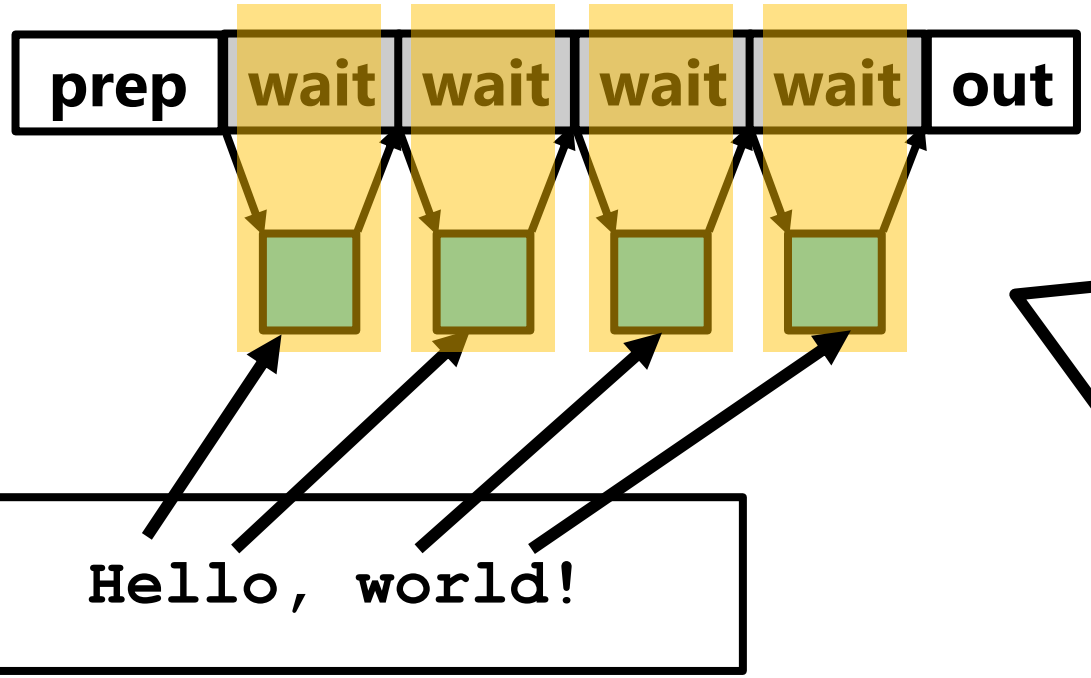

```
Hello, world!
```

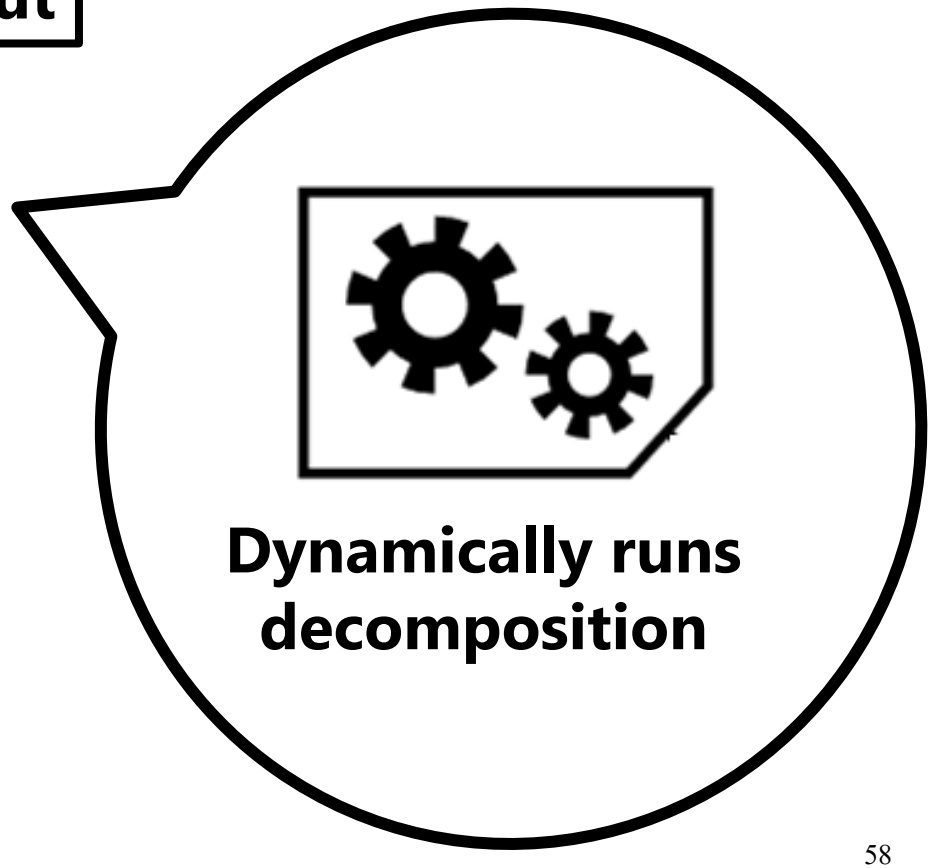


Solution: Peripheral Access Decomposition

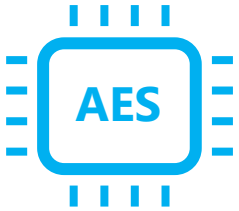


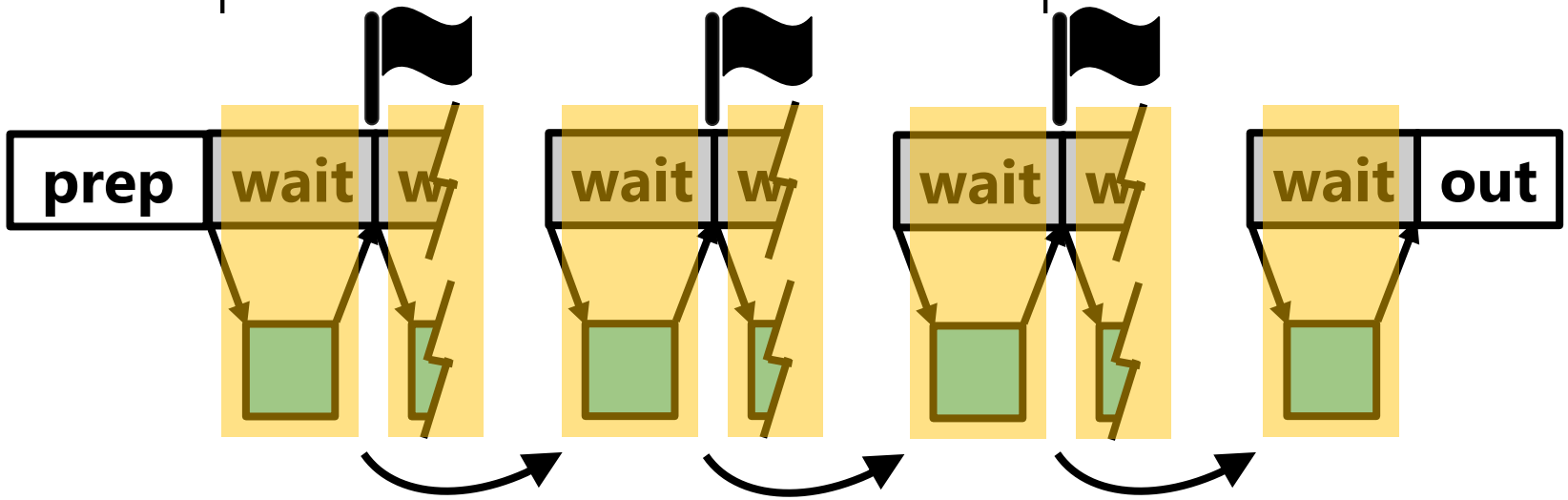


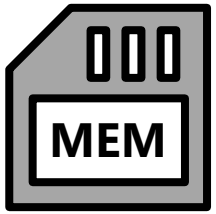




Solution: Peripheral Access Decomposition





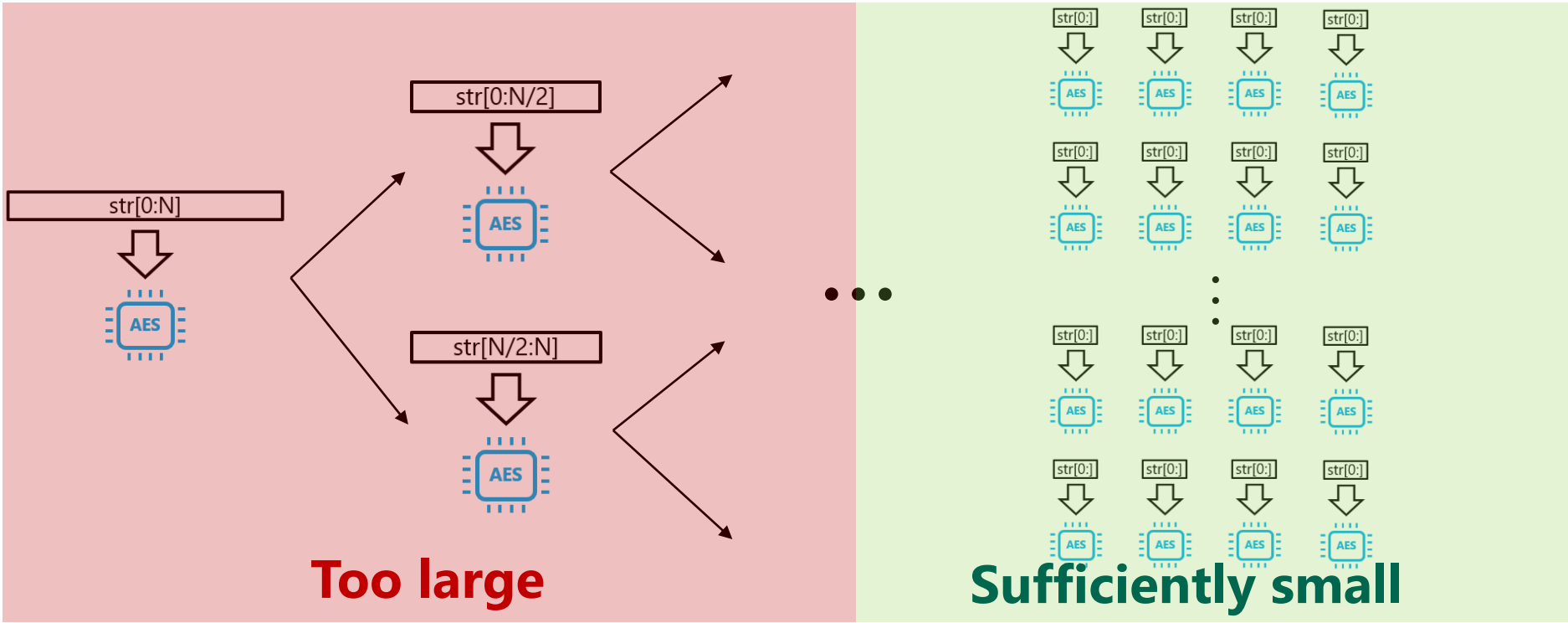



```
Hello, world!
```

Memoizes the appropriate size

Solution: Peripheral Access Decomposition

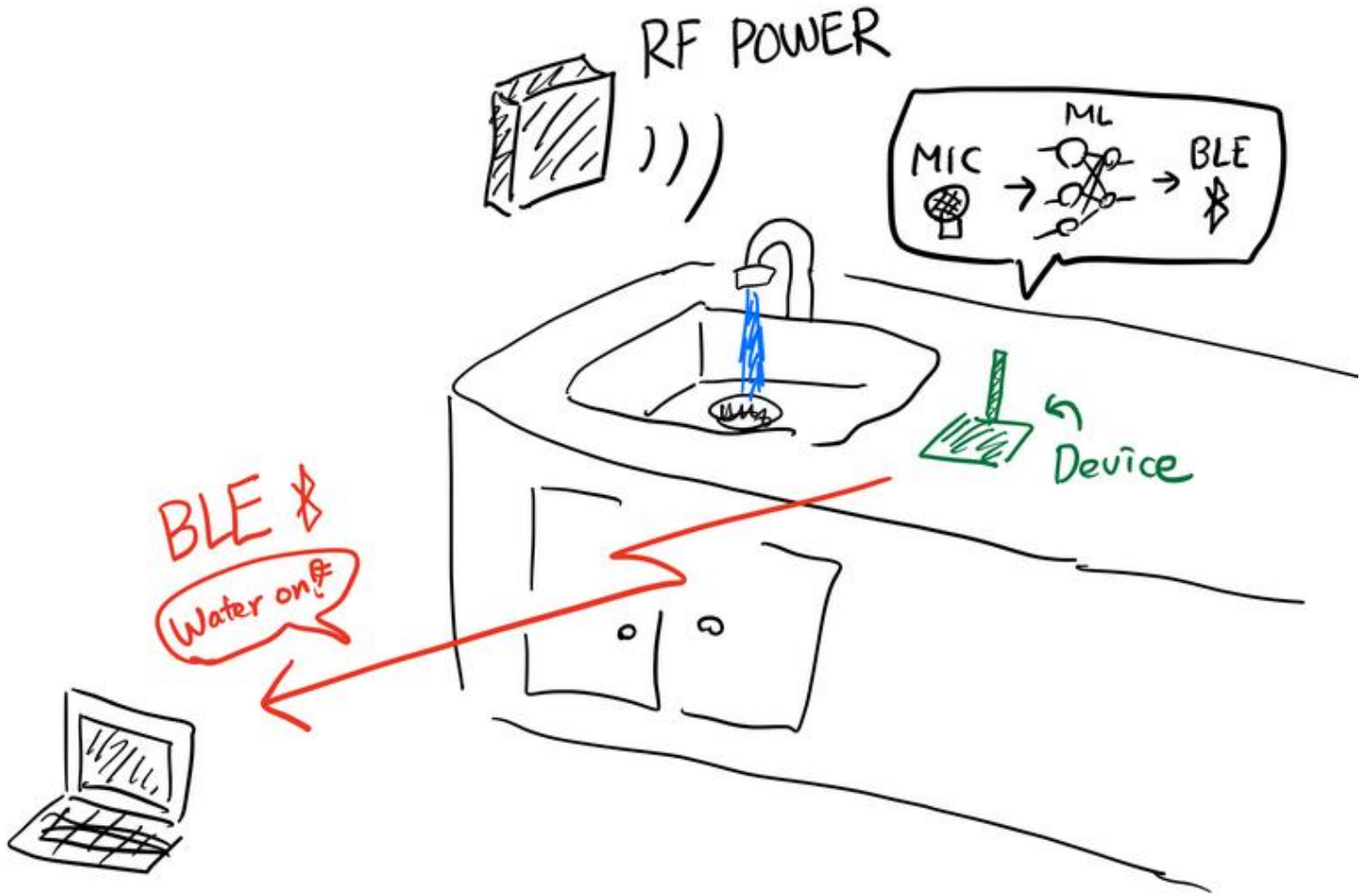
Check the energy use of the decomposed code



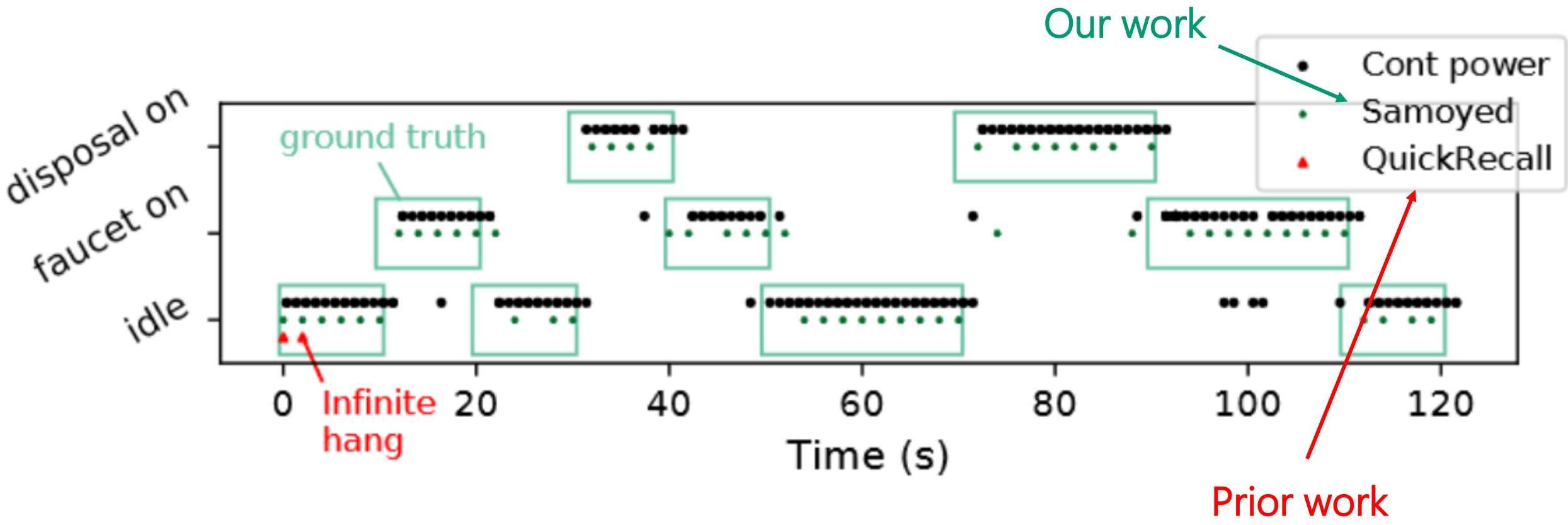
Evaluation: Kitchen Monitoring ML application

Send Bluetooth low-energy (BLE) alert if faucet or food dispenser is on.

Using microphone, ML accelerator, and BLE module.



Evaluation: Kitchen Monitoring ML application



Outline

- Challenge 1. Periodic Execution
- Solution 1. CatNap
- Challenge 2. Atomic Execution
- Solution 2. Samoyed
- **Future Work**

Summary and Future Work

- CatNap enables periodic execution with **recharge scheduling**, **feasibility test**, and **degradation**.
- Samoyed enables atomic execution with **dynamic region decomposition**.
- What is next?
  - What if there is not enough power at all (e.g., night)?
  - How fast/slow is the app going to be?
  - Can we do node-to-node communication?

System Challenges of Intermittent Computing

Kiwan Maeng