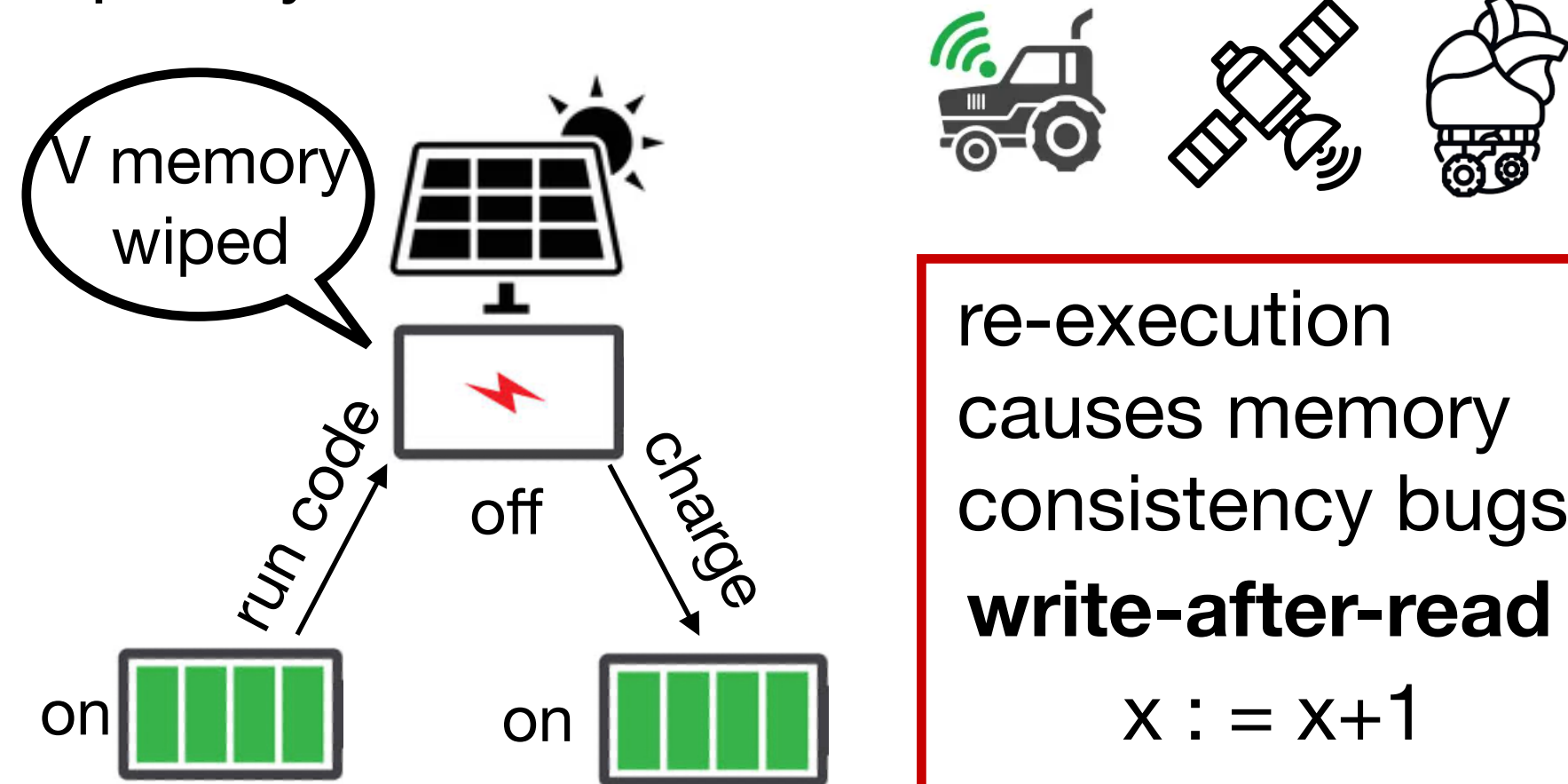


Intermittent Computing

batteryless energy-harvesting devices deployed in harsh or inaccessible environments crash frequently



correctness:

$\forall$ execution of a correct program,
 $\exists$ execution.

Concurrency: necessary but challenging

ID: sense; priority: 1
shared: {x}; policy: abort

1 x := 2;
2 ret

ID: main; priority: 2
shared: {x,y}
policy: re-execute

1 atom{
2 x := x + 1 };
3 print(y);
4 ret

ID: convert; priority: 3
shared: {x,y}; policy: abort

1 y := toFahren(x);
2 ret

hardware-triggered interrupts may not re-execute in the event of a power failure!

	x	y
NV	0	0
1	2	0
1	2	0
2	3	0
-	2	0
2	3	0
1	3	37.4
2	3	37.4
3	3	37.4
4	3	37.4

	x	y
NV	0	0
1	1	0
2	1	33.8
3	1	33.8
4	1	33.8

	x	y
NV	0	0
1	2	0
2	3	0
-	0	0
2	1	0
1	1	33.8
2	1	33.8
3	1	33.8
4	1	33.8

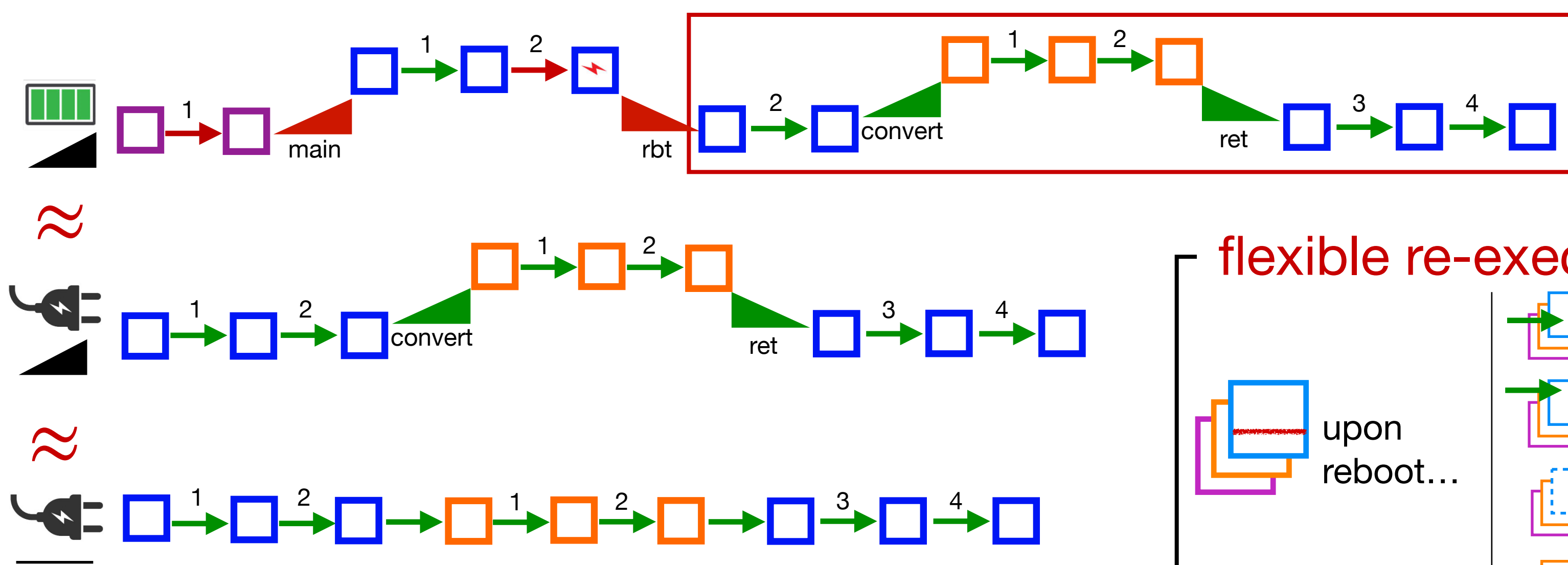
insufficient
checkpointing of x

need to checkpoint x in
sense but not in atom!

A calculus for correct concurrent, intermittent program execution

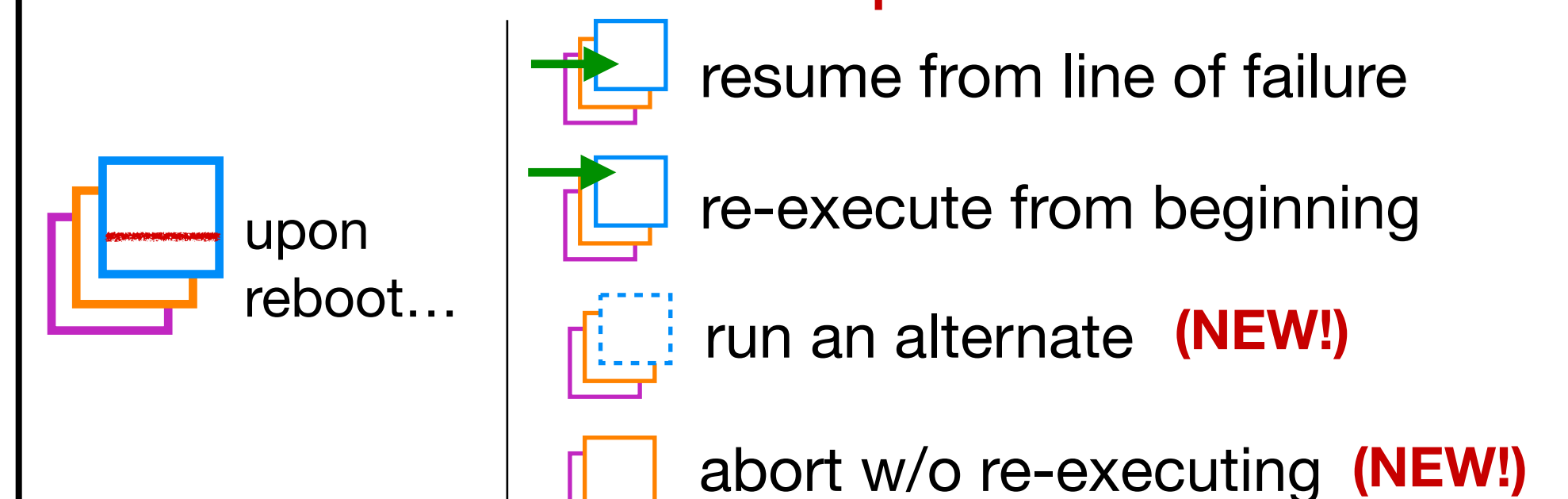
correctness

linearizability



**progress
& preservation**

flexible re-execution policies



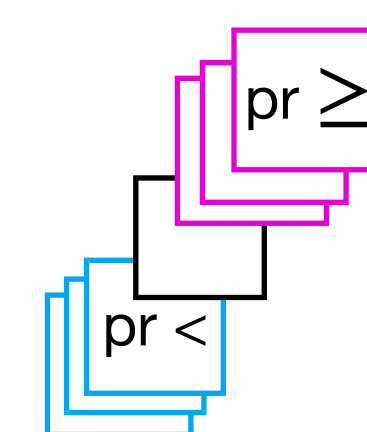
encoding variable (non-)idempotence as type qualifiers

is a value the **same (Id)** or **different (Nid)** on the next execution?

$\text{Id} \sqsubseteq \text{Nid}$

a variable's idempotence depends on how it is used **both within the task and by other tasks.**

$x : \text{int} @ \langle \text{Id}, \text{Id}, \text{Nid} \rangle$



check each task separately under a context with local and global effects

shared memory locations written by abort tasks behave as Nid unless checkpointed.

ID: sense; priority: 1
shared: {x}
policy: abort

1 x := 2;
2 ret

Γ_{sense}

$= \{x : \text{int} @ \langle \text{Id}, \text{Id}, \text{Id} \sqcup \text{Id} \rangle\}$

x is **checkpointed**

ID: main; priority: 2
shared: {x,y}
policy: re-execute

1 atom{
2 x := x + 1 };
3 print(y);
4 ret

$\Gamma_{\text{main}} = \{x : \text{int} @ \langle \text{Id}, \text{Id}, \text{Id} \rangle,$
 $y : \text{int} @ \langle \text{Id}, \text{Id}, \text{Nid} \rangle\}$

only read not shared w/ < task

ID: convert; priority: 3
shared: {x,y}
policy: abort

1 y := toFahren(x);
2 ret

Γ_{convert}

$= \{x : \text{int} @ \langle \text{Id}, \text{Id} \sqcup \text{Id}, \text{Id} \rangle,$
 $y : \text{int} @ \langle \text{Nid}, \text{Id}, \text{Id} \rangle\}$

y is written,
not checkpointed

not shared w/ ≥ task