

$$Asm_{thrd}(Syscall[tid, \varepsilon'_{cpu}, \varepsilon'_{thrd}]) \vdash \llbracket \mathbf{Ctx} \rrbracket$$

□

$$Asm_{mc}(Boot) \vdash \llbracket \mathbf{CertiKOS} \oplus \mathbf{Ctx} \rrbracket$$

Syscall ... █ █ ... █ ... █

⌞

⋮

TSched ... █ █ ... █ ... █

⌞

CSched (*Asm_{cpu}*) ... █ █

⌞

⋮

Boot (*Asm_{cpu}*) ... █ █

⌞

Boot (*Asm_{mc}*)

$$Asm_{thrd}(Syscall[tid, \varepsilon'_{cpu}, \varepsilon'_{thrd}]) \vdash \llbracket \mathbf{Ctxt} \rrbracket$$

⌞

$$Asm_{thrd}(TSched[tid, \varepsilon'_{cpu}, \varepsilon_{thrd}]) \vdash \llbracket \mathbf{CertiKOS}_{td} \oplus \mathbf{Ctxt} \rrbracket$$

⌞

$$Asm_{cpu}(CSched[cid, \varepsilon'_{cpu}]) \vdash \llbracket \mathbf{CertiKOS}_{td} \oplus \mathbf{Ctxt} \rrbracket$$

⌞

$$Asm_{cpu}(Boot[cid, \varepsilon_{cpu}]) \vdash \llbracket \mathbf{CertiKOS} \oplus \mathbf{Ctxt} \rrbracket$$

⌞

$$Asm_{mc}(Boot) \vdash \llbracket \mathbf{CertiKOS} \oplus \mathbf{Ctxt} \rrbracket$$

(where $\mathbf{CertiKOS} := \mathbf{CertiKOS}_{cpu} \oplus \mathbf{CertiKOS}_{td}$)

Boot (*Asm_{cpu}*) ... ■ ■

⌊

Boot (*Asm_{mc}*)

Asm_{cpu}(*Boot*[*cid*, *ε_{cpu}*]) ⊢ **[[CertiKOS ⊕ Ctxt]]**

⌊

Asm_{mc}(*Boot*) ⊢ **[[CertiKOS ⊕ Ctxt]]**

- Environment
 - Fixed number of CPUs
 - Fixed initial state for all CPUs
 - Fairness assumptions
- Things to solve
 - Hide non-determinism ①
 - Build environmental context for each CPU ②
 - Prove compositionality of multiple per-CPU machines ③
 - Provide simple environmental context for per-CPU machines ④

Optimize
environmental context

(4)

Introduce per-CPU machine (2)

Introduce partial machine (2, 3)
and prove linking theorem

Introduce hardware scheduler (1)

$$\begin{array}{c}
 Asm_{sep}(C, L[cid, \varepsilon_{sep}]) \vdash \llbracket \mathbf{Prog} \rrbracket \\
 \sqcup \\
 Asm_{reorder}(C, L[cid, \varepsilon'_{reorder}]) \vdash \llbracket \mathbf{Prog} \rrbracket \\
 \sqcup \\
 Asm_{reorder}(C, L[cid, \varepsilon_{reorder}]) \vdash \llbracket \mathbf{Prog} \rrbracket \\
 \sqcup \\
 Asm_{split}(C, L[cid, \varepsilon]) \vdash \llbracket \mathbf{Prog} \rrbracket \\
 \sqcup \\
 Asm_{big2}(C, L[cid, \varepsilon]) \vdash \llbracket \mathbf{Prog} \rrbracket \\
 \sqcup \\
 Asm_{big}(C, L[cid, \varepsilon]) \vdash \llbracket \mathbf{Prog} \rrbracket \\
 \sqcup \\
 Asm_{single}(C, L[cid, \varepsilon]) \vdash \llbracket \mathbf{Prog} \rrbracket \\
 \sqcup \\
 Asm_{env}(C, L[cid, \varepsilon]) \vdash \llbracket \mathbf{Prog} \rrbracket \\
 \sqcup \\
 Asm_{env}(C, \parallel_{i \in CoreSet} L[CoreSet, \varepsilon_{CoreSet}]) \vdash \llbracket \mathbf{Prog} \rrbracket \\
 \sqcup \\
 Asm_{oracle}(C, L[\varepsilon_{CoreSet}]) \vdash \llbracket \mathbf{Prog} \rrbracket \\
 \sqcup \\
 Asm_{mc}(C, L) \vdash \llbracket \mathbf{Prog} \rrbracket
 \end{array}$$

C : hardware configuration

L : an arbitrary layer with a certain condition

Link with Asm_{cpu} (4)

Optimize
environmental context

Introduce per-CPU machine (2)

Introduce partial machine (2, 3)
and prove linking theorem

Introduce hardware scheduler (1)

$$Asm_{cpu}(Boot[cid, \varepsilon_{cpu}]) \vdash \llbracket \text{CertiKOS} \oplus \text{Ctxt} \rrbracket$$

$$Asm_{sep}(Boot[cid, \varepsilon_{sep}]) \vdash \llbracket \text{CertiKOS} \oplus \text{Ctxt} \rrbracket$$

$$Asm_{reorder}(Boot[cid, \varepsilon'_{reorder}]) \vdash \llbracket \text{CertiKOS} \oplus \text{Ctxt} \rrbracket$$

$$Asm_{reorder}(Boot[cid, \varepsilon_{reorder}]) \vdash \llbracket \text{CertiKOS} \oplus \text{Ctxt} \rrbracket$$

$$Asm_{split}(Boot[cid, \varepsilon]) \vdash \llbracket \text{CertiKOS} \oplus \text{Ctxt} \rrbracket$$

$$Asm_{big2}(Boot[cid, \varepsilon]) \vdash \llbracket \text{CertiKOS} \oplus \text{Ctxt} \rrbracket$$

$$Asm_{big}(Boot[cid, \varepsilon]) \vdash \llbracket \text{CertiKOS} \oplus \text{Ctxt} \rrbracket$$

$$Asm_{single}(Boot[cid, \varepsilon]) \vdash \llbracket \text{CertiKOS} \oplus \text{Ctxt} \rrbracket$$

$$Asm_{env}(Boot[cid, \varepsilon]) \vdash \llbracket \text{CertiKOS} \oplus \text{Ctxt} \rrbracket$$

$$Asm_{env}(\parallel_{i \in \text{CoreSet}} Boot[\text{CoreSet}, \varepsilon_{\text{CoreSet}}]) \vdash \llbracket \text{CertiKOS} \oplus \text{Ctxt} \rrbracket$$

$$Asm_{oracle}(Boot[\varepsilon_{\text{CoreSet}}]) \vdash \llbracket \text{CertiKOS} \oplus \text{Ctxt} \rrbracket$$

$$Asm_{mc}(Boot) \vdash \llbracket \text{CertiKOS} \oplus \text{Ctxt} \rrbracket$$

TSched ...

⌞

CSched (*Asm_{cpu}*) ...

⌞

Boot (*Asm_{cpu}*) ...

⌞

Boot (*Asm_{mc}*)

$$Asm_{thrd}(TSched[tid, \varepsilon'_{cpu}, \varepsilon_{thrd}]) \vdash \llbracket \mathbf{CertiKOS}_{td} \oplus \mathbf{Ctxt} \rrbracket$$

⌞

$$Asm_{cpu}(CSched[cid, \varepsilon'_{cpu}]) \vdash \llbracket \mathbf{CertiKOS}_{td} \oplus \mathbf{Ctxt} \rrbracket$$

$$Asm_{cpu}(Boot[cid, \varepsilon_{ccpu}]) \vdash \llbracket \mathbf{CertiKOS} \oplus \mathbf{Ctxt} \rrbracket$$

⌞

$$Asm_{mc}(Boot) \vdash \llbracket \mathbf{CertiKOS} \oplus \mathbf{Ctxt} \rrbracket$$

(where $\mathbf{CertiKOS} := \mathbf{CertiKOS}_{cpu} \oplus \mathbf{CertiKOS}_{td}$)

$$TSched[tid, \varepsilon'_{cpu}, \varepsilon_{thrd}](yield) - (lst, log) \rightarrow (lst, log')$$

$$\boxed{TSched} \cdots \cdots \cdots \quad Asm_{thrd}(TSched[tid, \varepsilon'_{cpu}, \varepsilon_{thrd}]) \vdash \llbracket \mathbf{CertiKOS}_{td} \oplus \mathbf{Ctxt} \rrbracket$$

□□

□□

$$\boxed{CSched(Asm_{cpu})} \cdots \cdots \quad Asm_{cpu}(CSched[cid, \varepsilon'_{ci}]) \vdash \llbracket \mathbf{CertiKOS}_{td} \oplus \mathbf{Ctxt} \rrbracket$$

$CSched[cid, \varepsilon'_{ci}]$ contains software scheduler primitives

- spawn / yield / sleep / wakeup

$$CSched[cid, \varepsilon'_{ci}](yield) - (lst, log) \rightarrow (lst/[tid = \cdots, \rho = \cdots, \cdots], log')$$

- Environment
 - Arbitrary active or available thread set on the CPU
 - Dynamic initial states
- Things to solve
 - Hide context switching between threads ①
 - Build environmental context for each thread ②
 - Assign proper initial states for each thread ③
 - Prove compositionality of multiple per-thread machines ④
 - Use the same compiler for per-CPU/thread machines ⑤

Link per-CPU machine (5)
compiler with per-thread machine

Introduce (1, 2, 3)
per-thread machine

Introduce (1, 2, 3, 4)
multithreaded machine and
prove linking theorem

$$Asm_{thrd}(C, TSched[tid, \varepsilon'_{cpu}, \varepsilon_{thrd}]) \vdash \llbracket \mathbf{CertiKOS}_{td} \oplus \mathbf{Ctx} \rrbracket$$

\sqcup

$$IAsm_{thrd}(C, TLink[tid, \varepsilon'_{cpu}, \varepsilon_T^{zip}]) \vdash \llbracket \mathbf{CertiKOS}_{td} \oplus \mathbf{Ctx} \rrbracket$$

\sqcup

$$IAsm_{mt}(C, TLink[tid, \varepsilon'_{cpu}, \varepsilon_T]) \vdash \llbracket \mathbf{CertiKOS}_{td} \oplus \mathbf{Ctx} \rrbracket$$

\sqcup

$$IAsm_{mt}(C, \parallel_{ti \in TSet} TLink[cid, \varepsilon'_{cpu}]) \vdash \llbracket \mathbf{CertiKOS}_{td} \oplus \mathbf{Ctx} \rrbracket$$

\sqcup

$$Asm_{mt}(C, \parallel_{ti \in TSet} TLink[cid, \varepsilon'_{cpu}]) \vdash \llbracket \mathbf{CertiKOS}_{td} \oplus \mathbf{Ctx} \rrbracket$$

\sqcup

$$Asm_{cpu}(C, CSched[cid, \varepsilon'_{cpu}]) \vdash \llbracket \mathbf{CertiKOS}_{td} \oplus \mathbf{Ctx} \rrbracket$$

C : thread configuration

abstract relations
between two layers

\longrightarrow

$AbsRelC$

\longrightarrow

$AbsRelT$

$CSched$: arbitrary layer with scheduling primitives
(context switching incl.)

$TLink$: arbitrary layer for intermediate machines
(scheduling primitives are defined in the machine itself)

$TSched$: arbitrary layer with scheduling primitives
(Scheduling has a identity behavior)

Link per-CPU machine (5)
compiler with per-thread machine

Introduce (1, 2, 3)
per-thread machine

Introduce (1, 2, 3, 4)
multithreaded machine and
prove linking theorem

$$Asm_{thrd}(PThread[tid, \varepsilon'_{cpu}, \varepsilon_{thrd}]) \vdash \llbracket \text{CertiKOS}_{td} \oplus \text{Ctx} \rrbracket$$

\sqcup

$$IAsm_{thrd}(PHBThread[tid, \varepsilon'_{cpu}, \varepsilon_T^{zip}]) \vdash \llbracket \text{CertiKOS}_{td} \oplus \text{Ctx} \rrbracket$$

\sqcup

$$IAsm_{mt}(PHBThread[tid, \varepsilon'_{cpu}, \varepsilon_T]) \vdash \llbracket \text{CertiKOS}_{td} \oplus \text{Ctx} \rrbracket$$

\sqcup

$$IAsm_{mt}(\parallel_{ti \in TSet} PHBThread[cid, \varepsilon'_{cpu}]) \vdash \llbracket \text{CertiKOS}_{td} \oplus \text{Ctx} \rrbracket$$

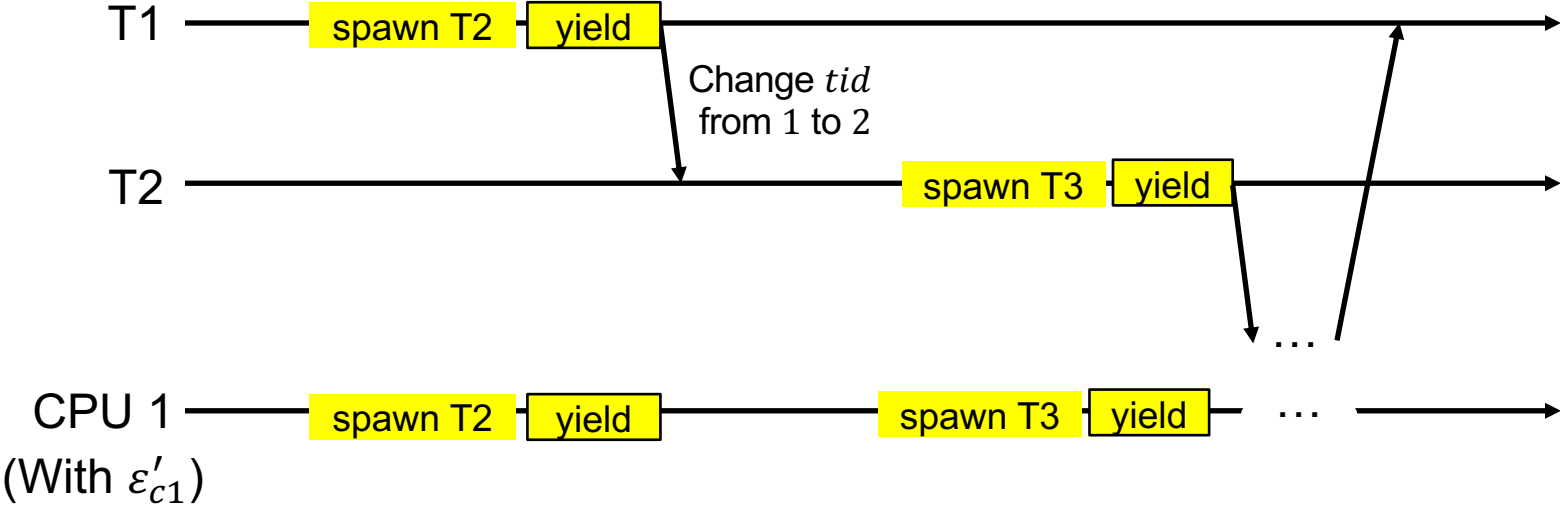
\sqcup

$$Asm_{mt}(\parallel_{ti \in TSet} PHBThread[cid, \varepsilon'_{cpu}]) \vdash \llbracket \text{CertiKOS}_{td} \oplus \text{Ctx} \rrbracket$$

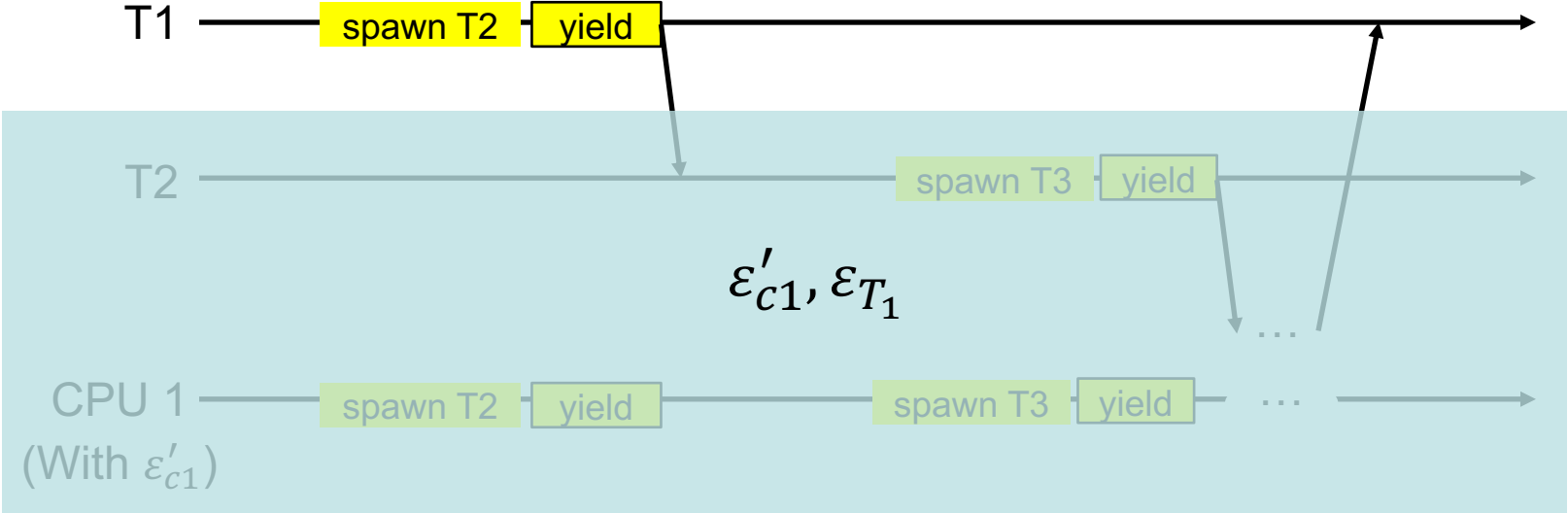
\sqcup

$$Asm_{cpu}(PThread[cid, \varepsilon'_{cpu}]) \vdash \llbracket \text{CertiKOS}_{td} \oplus \text{Ctx} \rrbracket$$

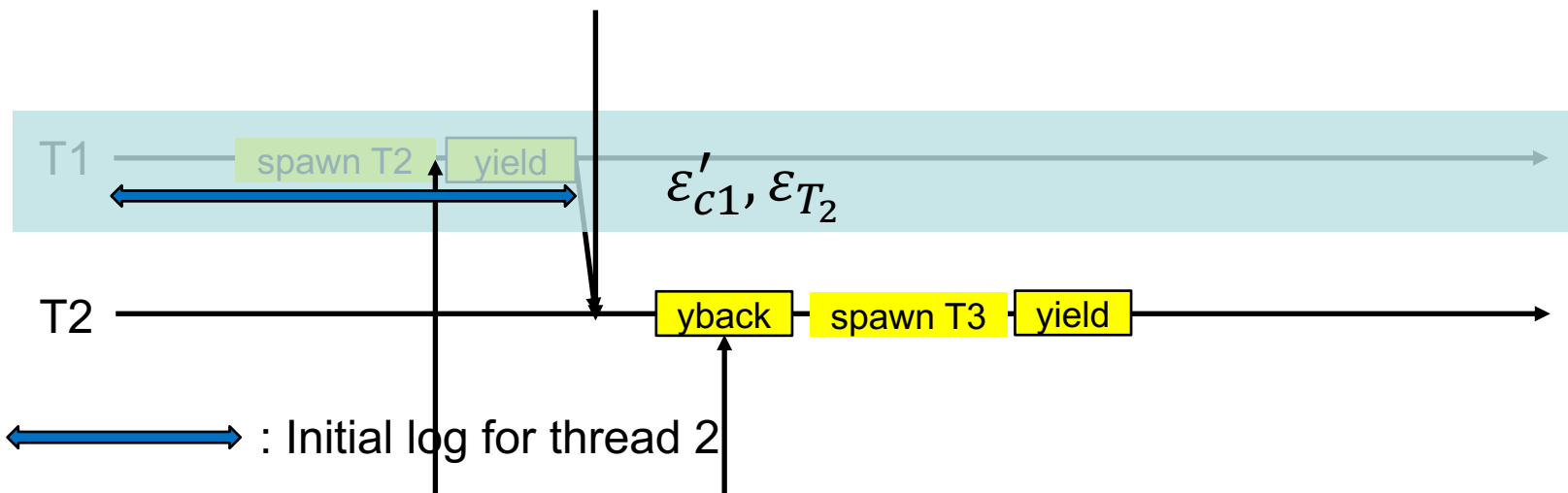
$$st_{TSet} := (tid, \{ti \mapsto lst_{ti}\}, log) \ (\forall ti, ti \in TSet)$$



$$st_{TSet} := (tid, \{ti \mapsto lst_{ti}\}, log) \ (\forall ti, ti \in TSet)$$



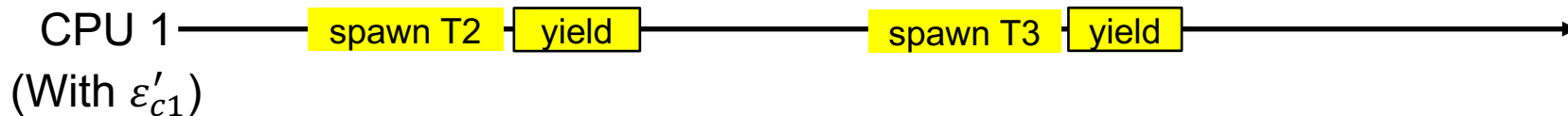
$T_{\text{status}}(1) = (\text{Run}, \text{Active})$
 $T_{\text{status}}(2) = (\text{Ready}, \text{Inactive})$

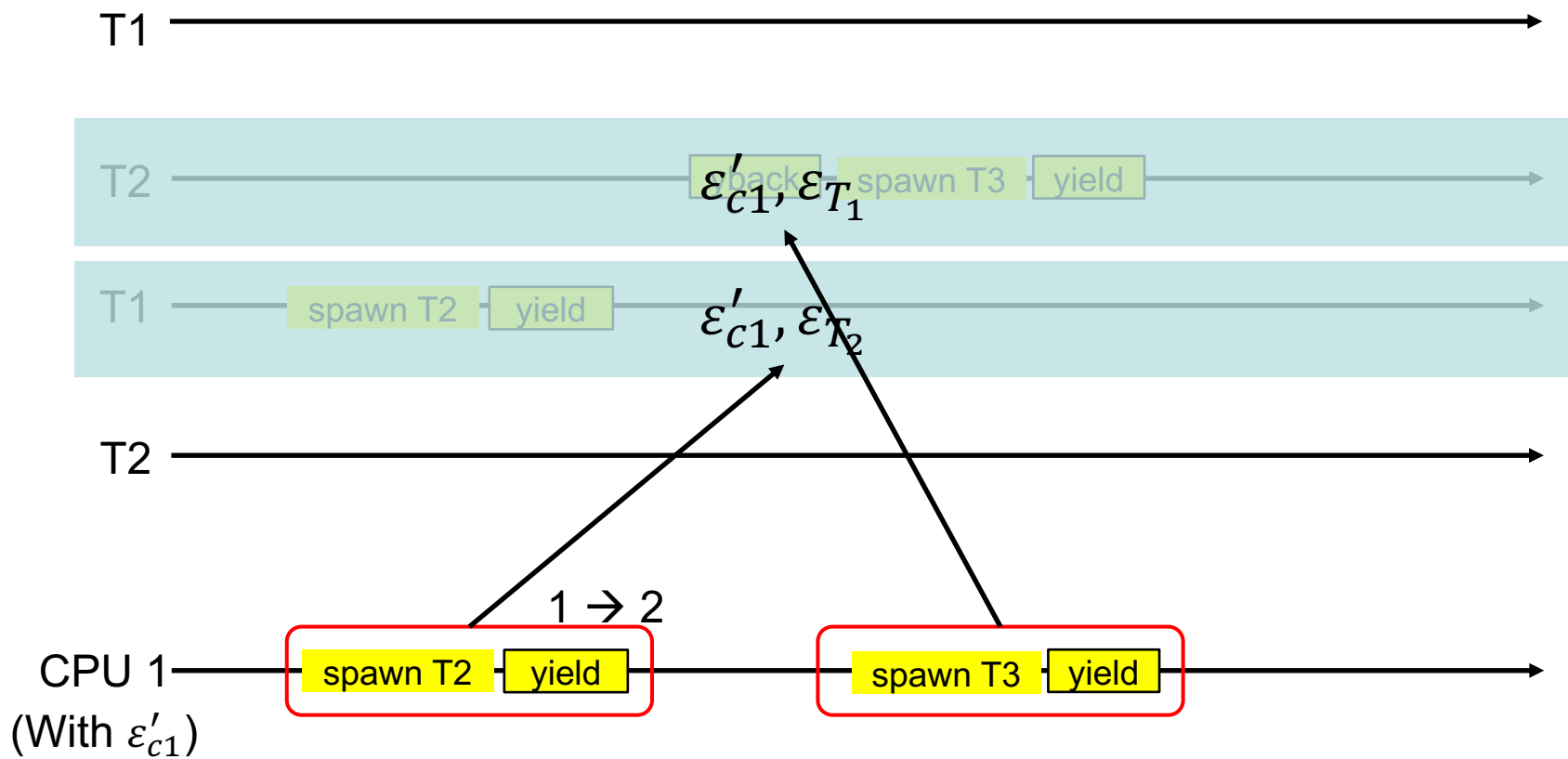


$T_{\text{status}}(1) = (\text{Ready}, \text{Active})$
 $T_{\text{status}}(2) = (\text{Run}, \text{Inactive})$

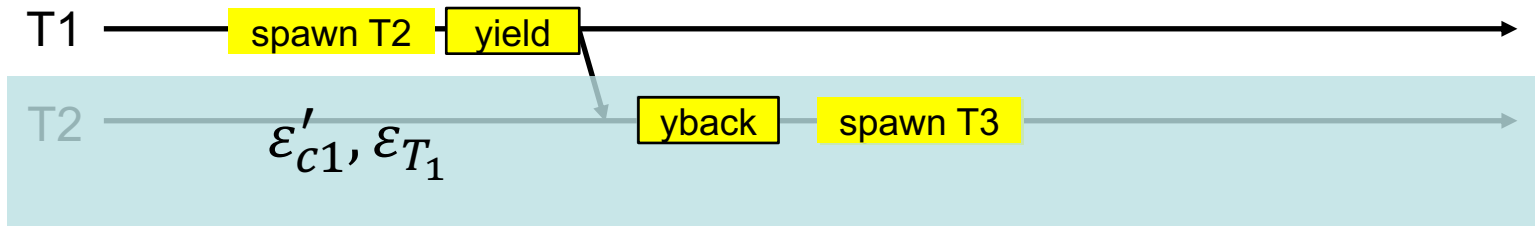
$T_{\text{status}}(1) = (\text{Ready}, \text{active})$
 $T_{\text{status}}(2) = (\text{Run}, \text{active})$

$1 \rightarrow 2$



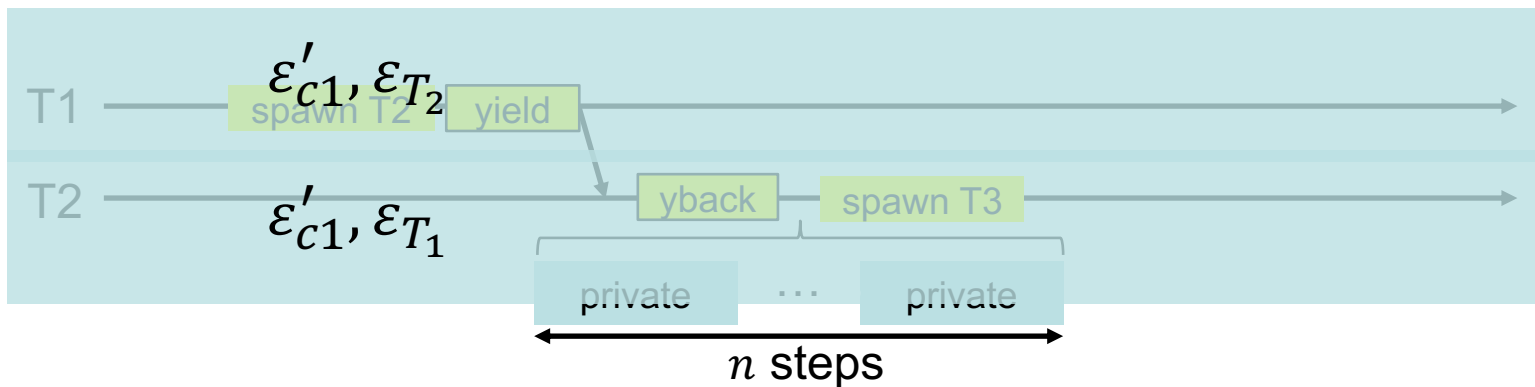


$IAsm_{mt}$
with T1



$n < \text{progress}$
Every thread will generate **at least one event** within progress steps

$IAsm_{mt}$
with TSet



Initial state: Calculate initial log to find the proper initial state

Yield rule: $TSched[tid, \varepsilon'_{cpu}, \varepsilon_{thrd}](yield) - (lst, log) \rightarrow (lst, log')$

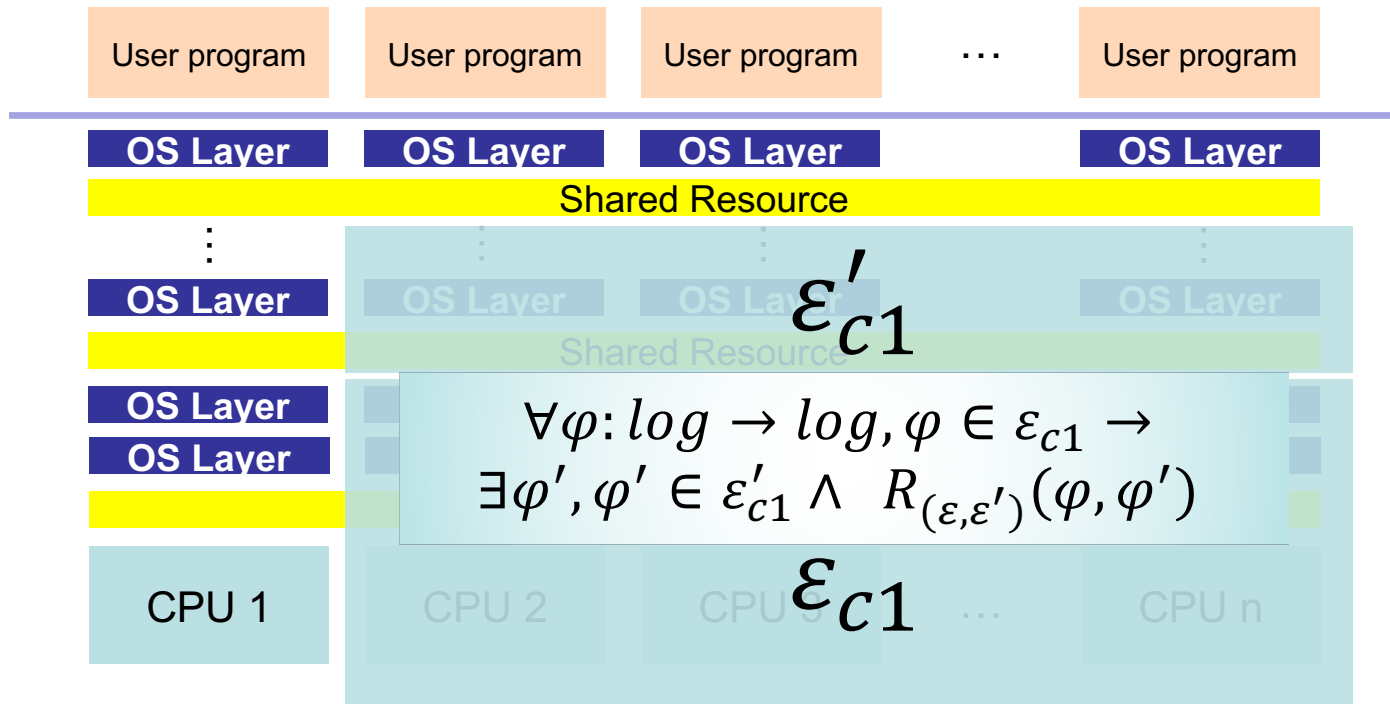
$$\begin{array}{c}
 \boxed{TSched} \cdots \boxed{} \cdots \boxed{} \cdots \boxed{} \cdots \boxed{} \quad Asm_{thrd}(TSched[tid, \varepsilon'_{cpu}, \varepsilon_{thrd}]) \vdash \llbracket \mathbf{CertiKOS}_{td} \oplus \mathbf{Ctxt} \rrbracket \\
 \sqcup \quad \quad \quad \sqcup
 \end{array}$$

$$\boxed{CSched(Asm_{cpu})} \cdots \boxed{} \cdots \boxed{} \quad Asm_{cpu}(CSched[tid, \varepsilon'_{cpu}]) \vdash \llbracket \mathbf{CertiKOS}_{td} \oplus \mathbf{Ctxt} \rrbracket$$

Initial state: Fixed initial state

Yield rule: $CSched[cid, \varepsilon'_{cpu}](yield) - (lst, log) \rightarrow (lst/[tid = \cdots, \rho = \cdots, \cdots], log')$

Environmental Context Relation



Hide Nondeterminism

$$Asm_{oracle}(Boot[\varepsilon_{CoreSet}]) \vdash \llbracket \mathbf{CertiKOS} \oplus \mathbf{Ctx} \rrbracket$$

\sqcup

$$Asm_{mc}(Boot) \vdash \llbracket \mathbf{CertiKOS} \oplus \mathbf{Ctx} \rrbracket$$