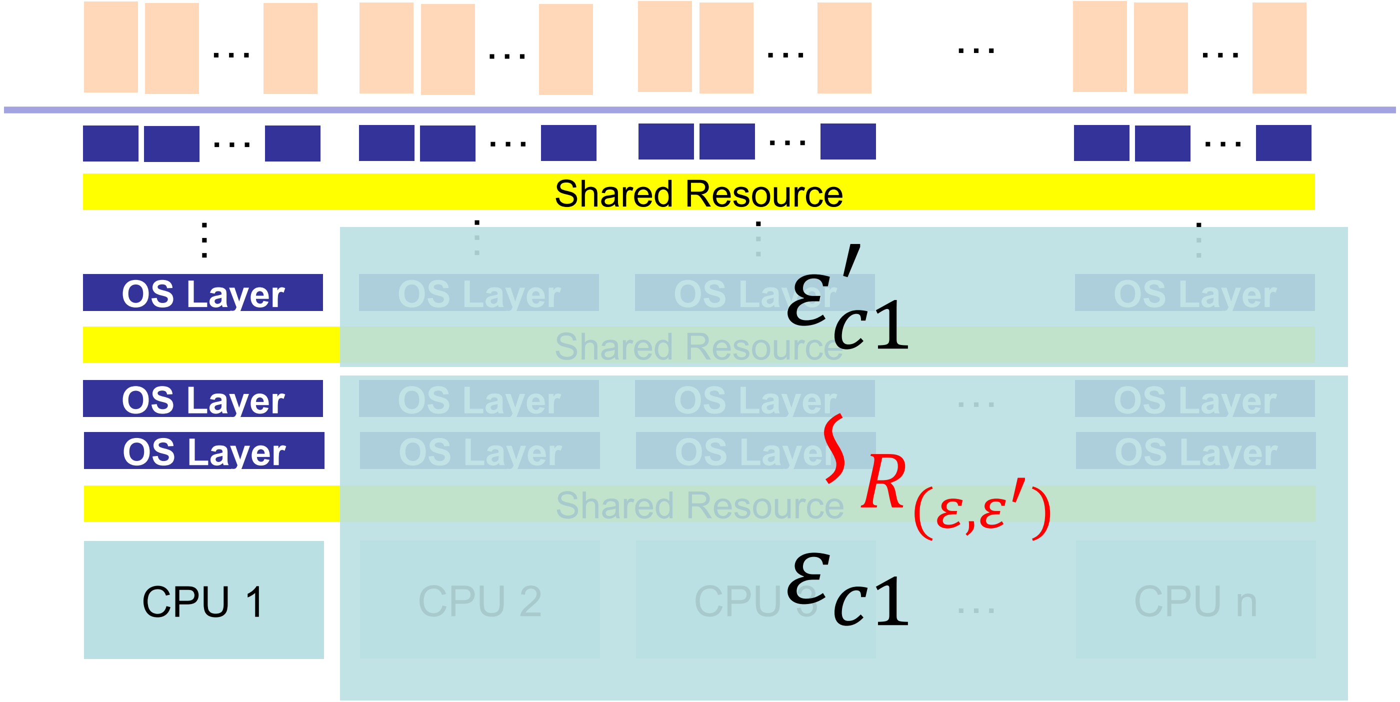

Shared Resource
OS Layer
OS Layer
OS Layer
OS Layer
Shared Resource
ε'_{c1}
OS Layer
OS Layer
OS Layer
OS Layer
OS Layer
OS Layer
OS Layer
OS Layer
OS Layer
$\int_{R_{(\varepsilon,\varepsilon')}}$
Shared Resource
Shared Resource
ε_{c1}
CPU 1
CPU 2
CPU 3
CPU n

(5) Connect local layer interface

$\mathrm{Mach}_{\mathrm{LAsm}}(C, L[cid, \varepsilon_{\mathrm{cpu}}]) \vdash [\![\mathbf{Prog}]\!]$

$\sqsubseteq$

(4) Optimize environmental context

$\mathrm{Mach}_{sep}(C, L[cid, \varepsilon_{sep}]) \vdash [\![\mathbf{Prog}]\!]$

$\sqsubseteq$

$\mathrm{Mach}_{\mathrm{reorder}}(C, L[cid, \varepsilon'_{reorder}]) \vdash [\![\mathbf{Prog}]\!]$

$\sqsubseteq$

$\mathrm{Mach}_{\mathrm{reorder}}(C, L[cid, \varepsilon_{reorder}]) \vdash [\![\mathbf{Prog}]\!]$

$\sqsubseteq$

$\mathrm{Mach}_{\mathrm{split}}(C, L[cid, \varepsilon]) \vdash [\![\mathbf{Prog}]\!]$

$\sqsubseteq$

$\mathrm{Mach}_{\mathrm{si\_big'}}(C, L[cid, \varepsilon]) \vdash [\![\mathbf{Prog}]\!]$

$\sqsubseteq$

$\mathrm{Mach}_{\mathrm{si\_big}}(C, L[cid, \varepsilon]) \vdash [\![\mathbf{Prog}]\!]$

$\sqsubseteq$

$\mathrm{Mach}_{\mathrm{si}}(C, L[cid, \varepsilon]) \vdash [\![\mathbf{Prog}]\!]$

(2) Introduce per-CPU machine

$\sqsubseteq$

$\mathrm{Mach}_{\mathrm{env}}(C, L[cid, \varepsilon]) \vdash [\![\mathbf{Prog}]\!]$

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

$\sqsubseteq$

$\mathrm{Mach}_{\mathrm{env}}(C, \|_{i \in CoreSet}\ L[CoreSet, \varepsilon_{CoreSet}]) \vdash [\![\mathbf{Prog}]\!]$

$\sqsubseteq$

$\mathrm{Mach}_{\mathrm{oracle}}(C, L[\varepsilon_{CoreSet}]) \vdash [\![\mathbf{Prog}]\!]$

(1) Introduce hardware scheduler

$\sqsubseteq$

$\mathrm{Mach}_{\mathrm{mc}}(C, L) \vdash [\![\mathbf{Prog}]\!]$

C: hardware configuration L: an arbitrary layer with a certain condition

$\text{Mach}_{\text{HAsm}}(C, \text{TSched}[tid, \varepsilon_{tid}]) \vdash [\![\textbf{Prog}]\!]$

(5) Connect CompCertX Interface

$\sqsubseteq$

$\text{Mach}_{\text{TAsm}}(C, \text{TLink}[tid, \varepsilon_{tid}^{MTLink}]) \vdash [\![\textbf{Prog}]\!]$

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

$\sqsubseteq$

$\text{Mach}_{\text{IEAsm}}(C, \text{TLink}[tid, \varepsilon_{tid}^{MTLink}]) \vdash [\![\textbf{Prog}]\!]$

$\sqsubseteq$

$\text{Mach}_{\text{IEAsm}}(\text{C}, \parallel tid \in T_{[cid]} \text{TLink}[cid, \varepsilon_{tid}^{MTLink}]) \vdash [\![\textbf{Prog}]\!]$

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

$\sqsubseteq$

$\text{Mach}_{\text{EAsm}}(\text{C}, \parallel tid \in T_{[cid]} \text{TLink}[cid, \varepsilon_{cid}]) \vdash [\![\textbf{Prog}]\!]$

$\sqsubseteq$

$\text{Mach}_{\text{LAsm}}(C, \text{CSched}[cid, \varepsilon_{cid}]) \vdash [\![\textbf{Prog}]\!]$

C: thread configuration

abstract relations between two layers

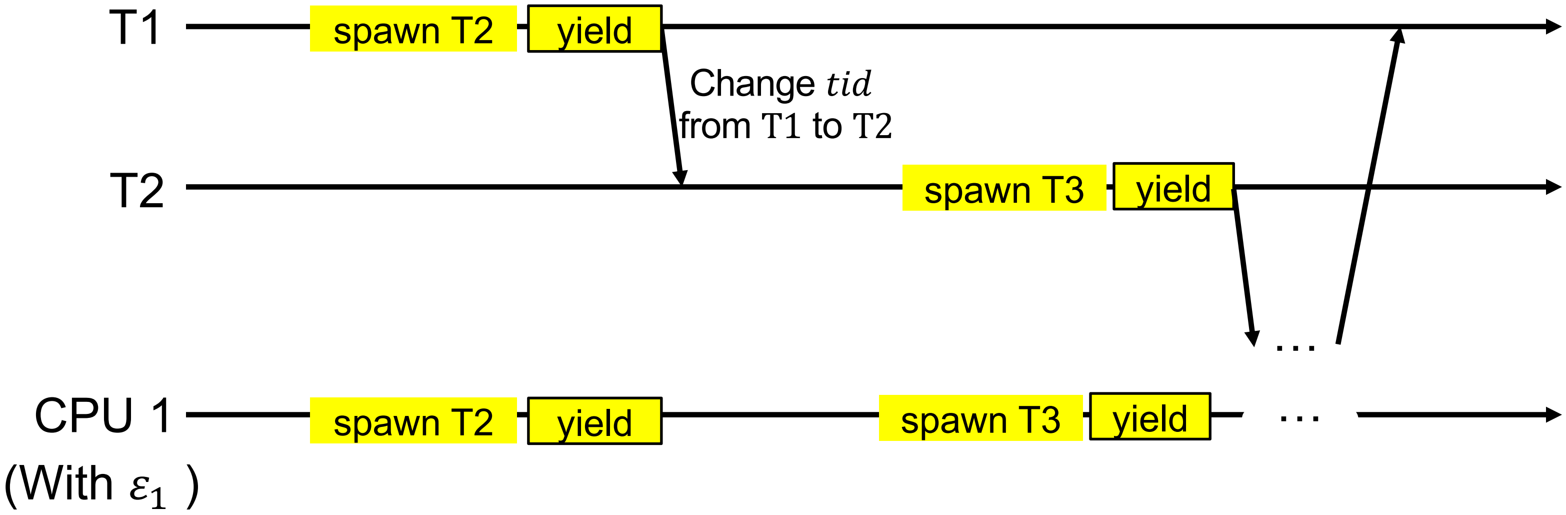
→ AbsRelC

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

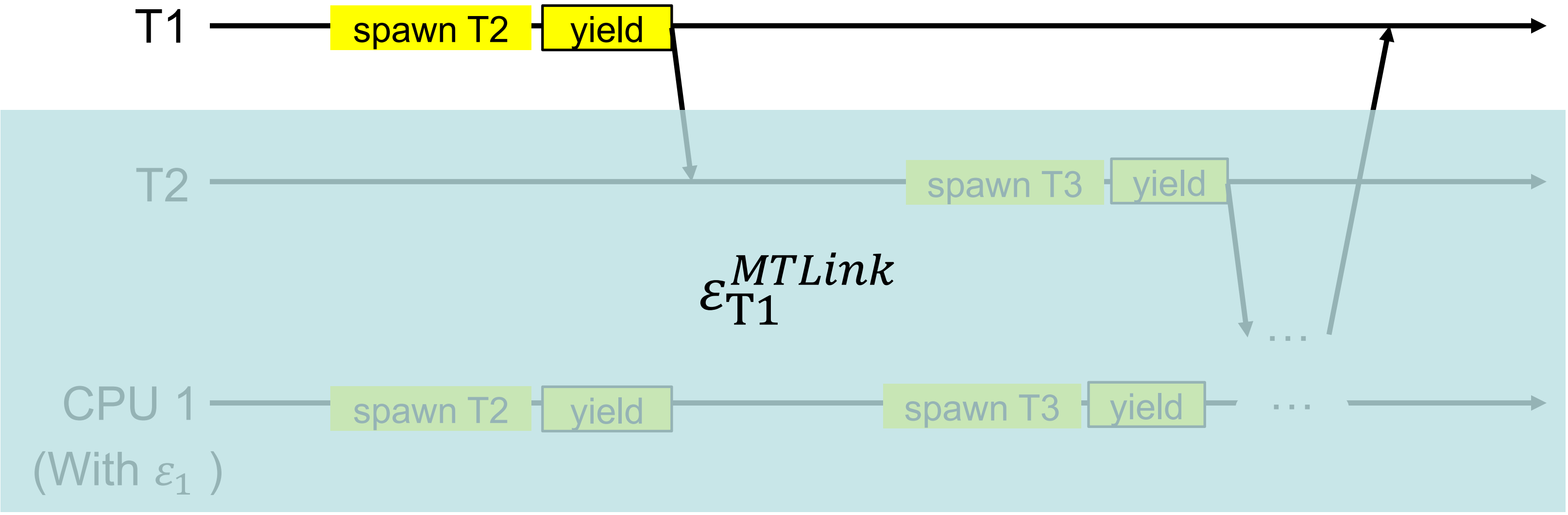
→ AbsRelT

- 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)

$\mathrm{estate_{mt}} \coloneqq (tid, mem, \{ti \mapsto lst_{ti}\}, log)\ (\forall ti, ti \in T_{[cid]})$



$$\text{estate}_{\text{mt}} \coloneqq (tid, mem, \{ti \mapsto lst_{ti}\}, log)\ (\forall ti, ti \in T_{[cid]})$$



T1

spawn T2

yield

T2

spawn T3

yield

$\varepsilon_{\text{T1}}^{MTLink}$

…

CPU 1

(With ε_1)

spawn T2

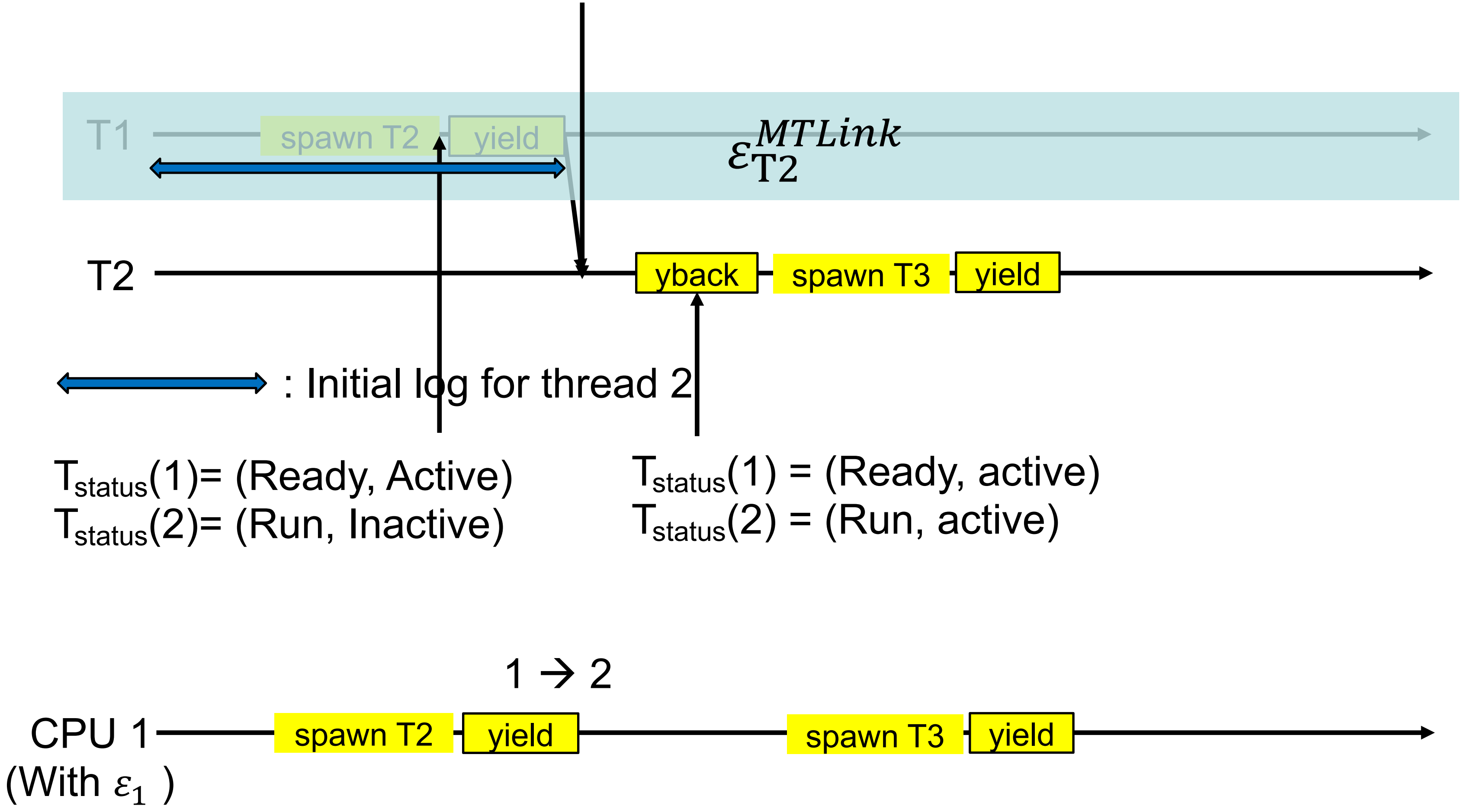
yield

spawn T3

yield

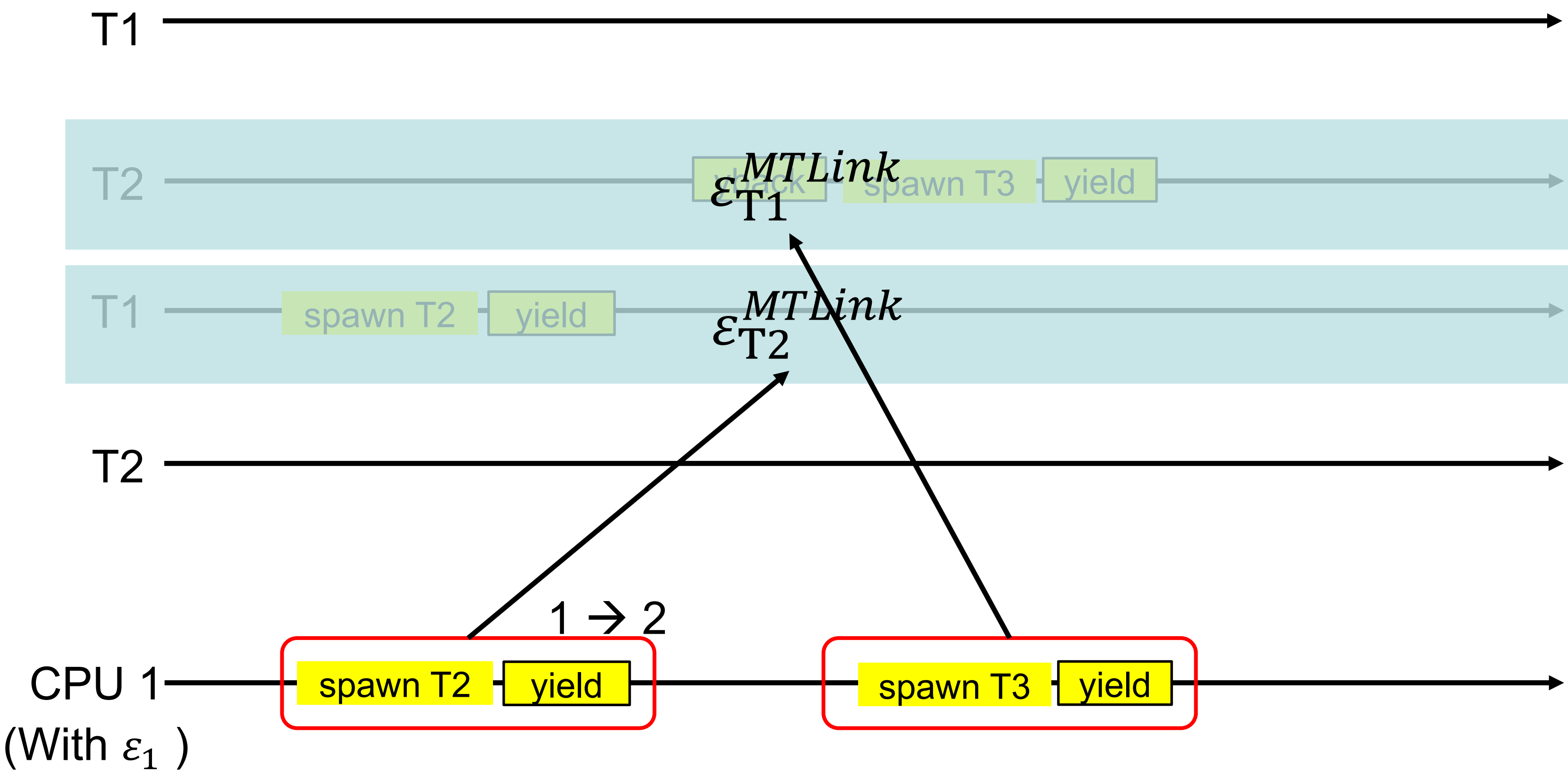
…

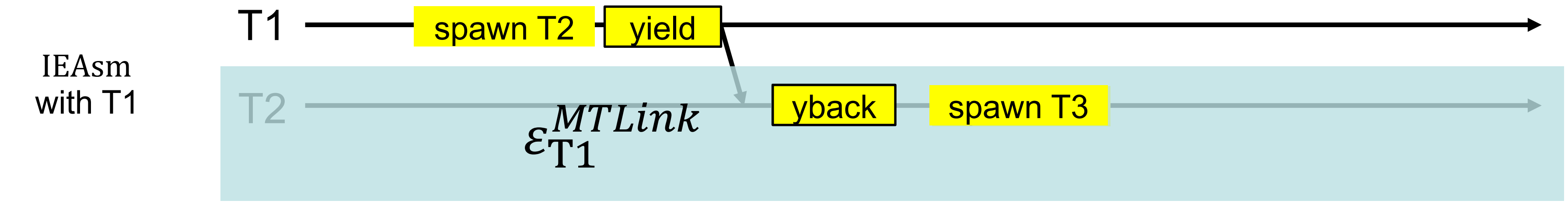
$T_{status}(1) = (Run, Active)$
$T_{status}(2) = (Ready, Inactive)$



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

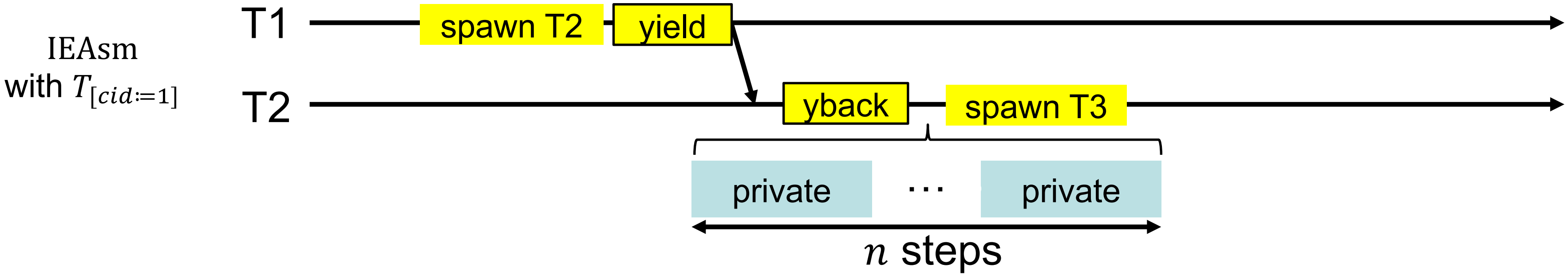
$T_{status}(1) = (Ready, active)$
$T_{status}(2) = (Run, active)$

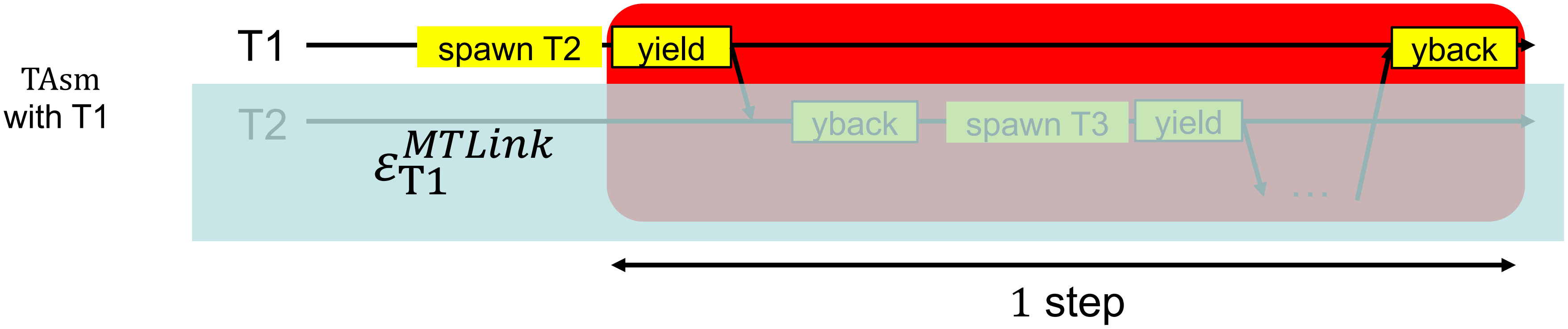

T1
T2
yback
spawn T3
yield
$\varepsilon_{\mathrm{T1}}^{MTLink}$
T1
spawn T2
yield
$\varepsilon_{\mathrm{T2}}^{MTLink}$
T2
1 → 2
CPU 1
(With ε_1)
spawn T2
yield
spawn T3
yield



$n < progress$

Every thread will generate at least one event within $progress$ *steps*





TAsm with T1

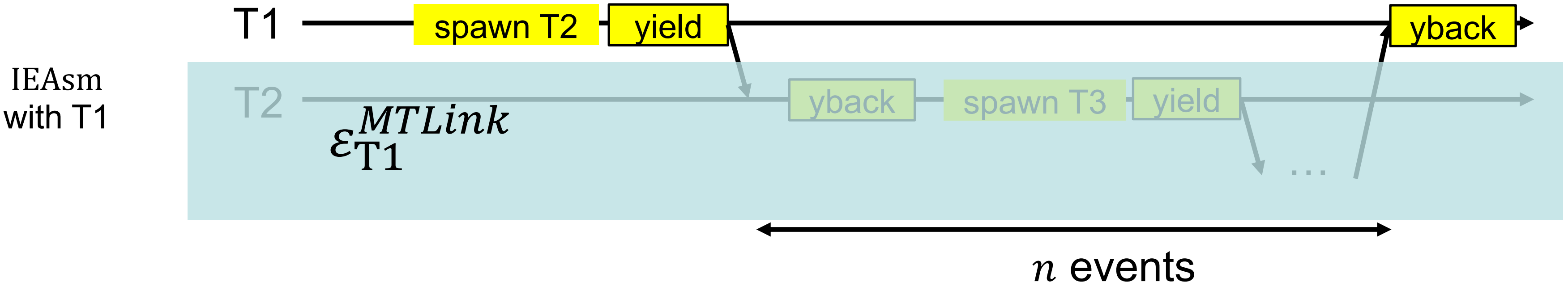
T1 spawn T2 yield yback

T2 $\mathcal{E}_{\mathrm{T1}}^{MTLink}$ yback spawn T3 yield …

1 step

$n \leq limit$

Every thread will be eventually scheduled within $limit \times progress$ steps



IEAsm with T1

T1 spawn T2 yield yback

T2 $\mathcal{E}_{\mathrm{T1}}^{MTLink}$ yback spawn T3 yield …

n events