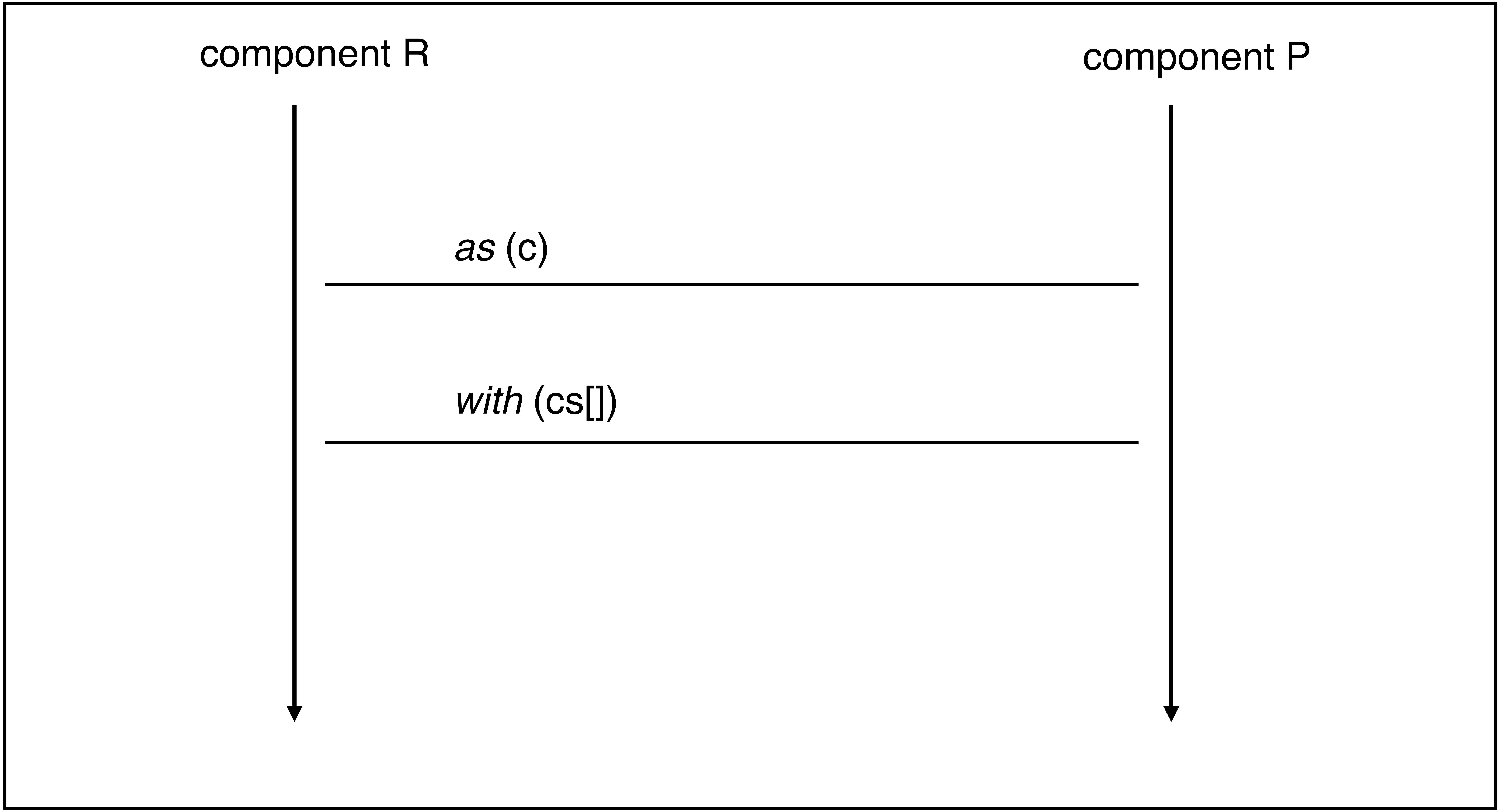
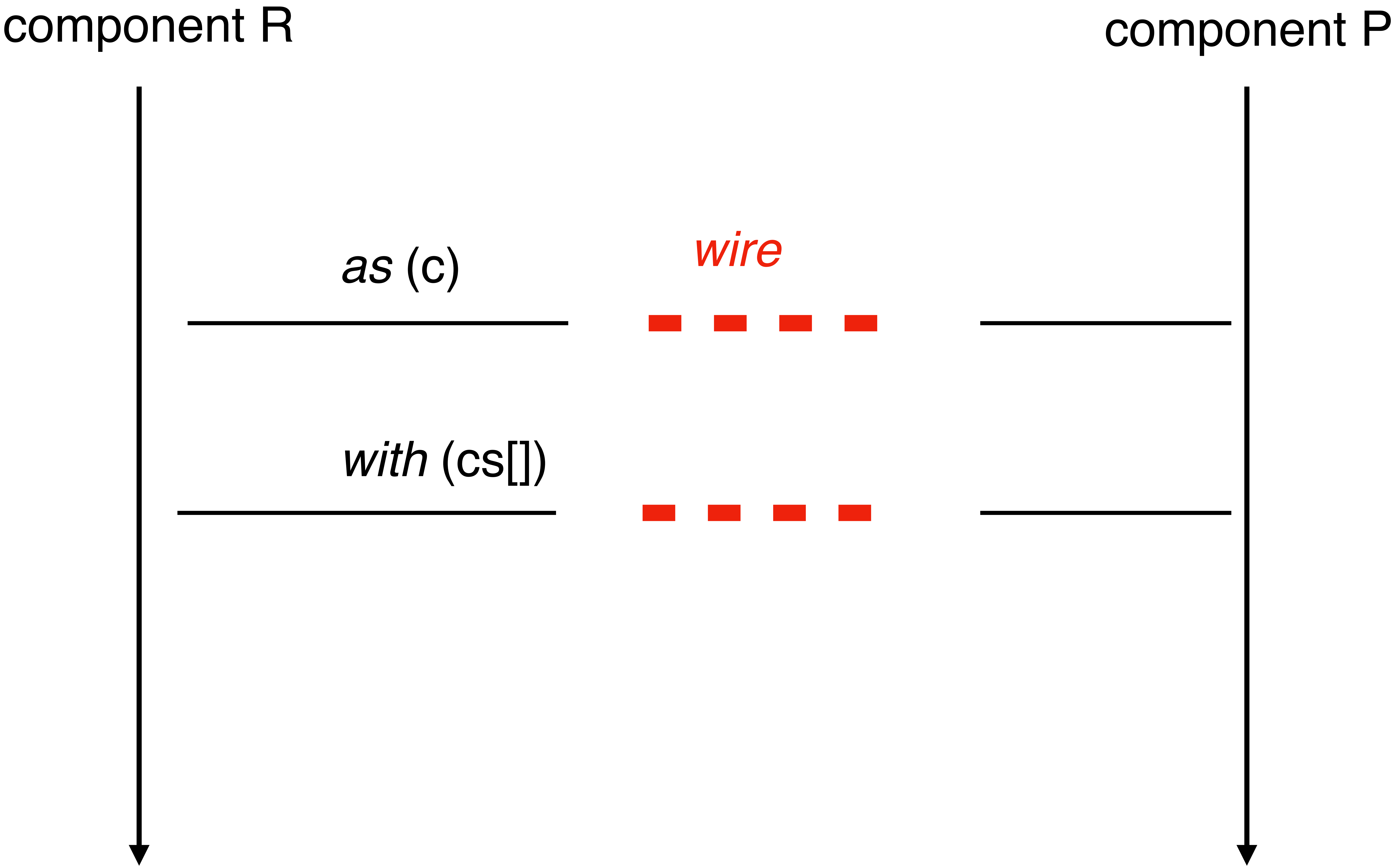
Adapting Existing Components to New Interfaces: Logical, Remote


component R
component P
as (c)
with (cs[])



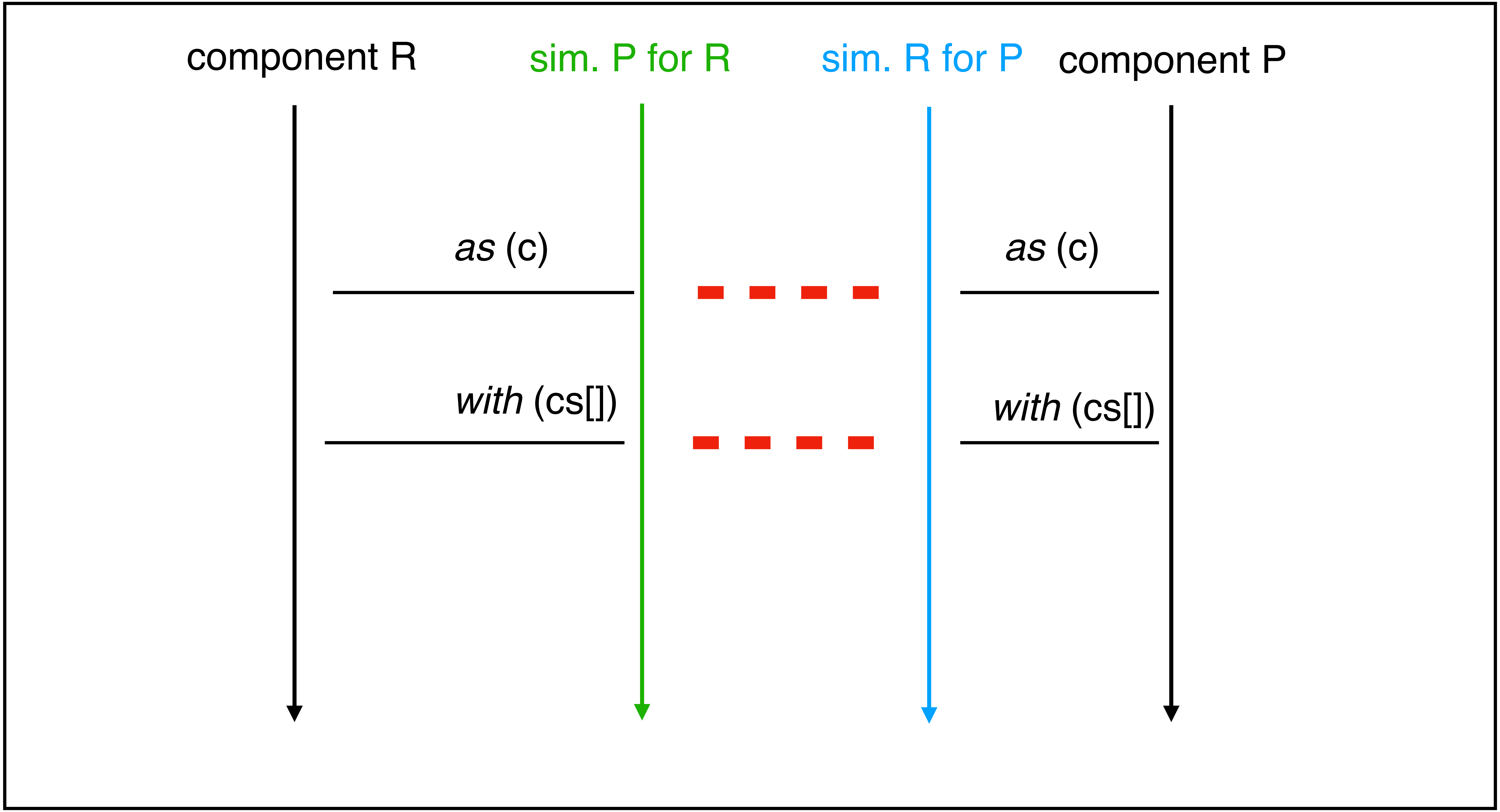
component R

component P

as (c)

wire

with (cs[])

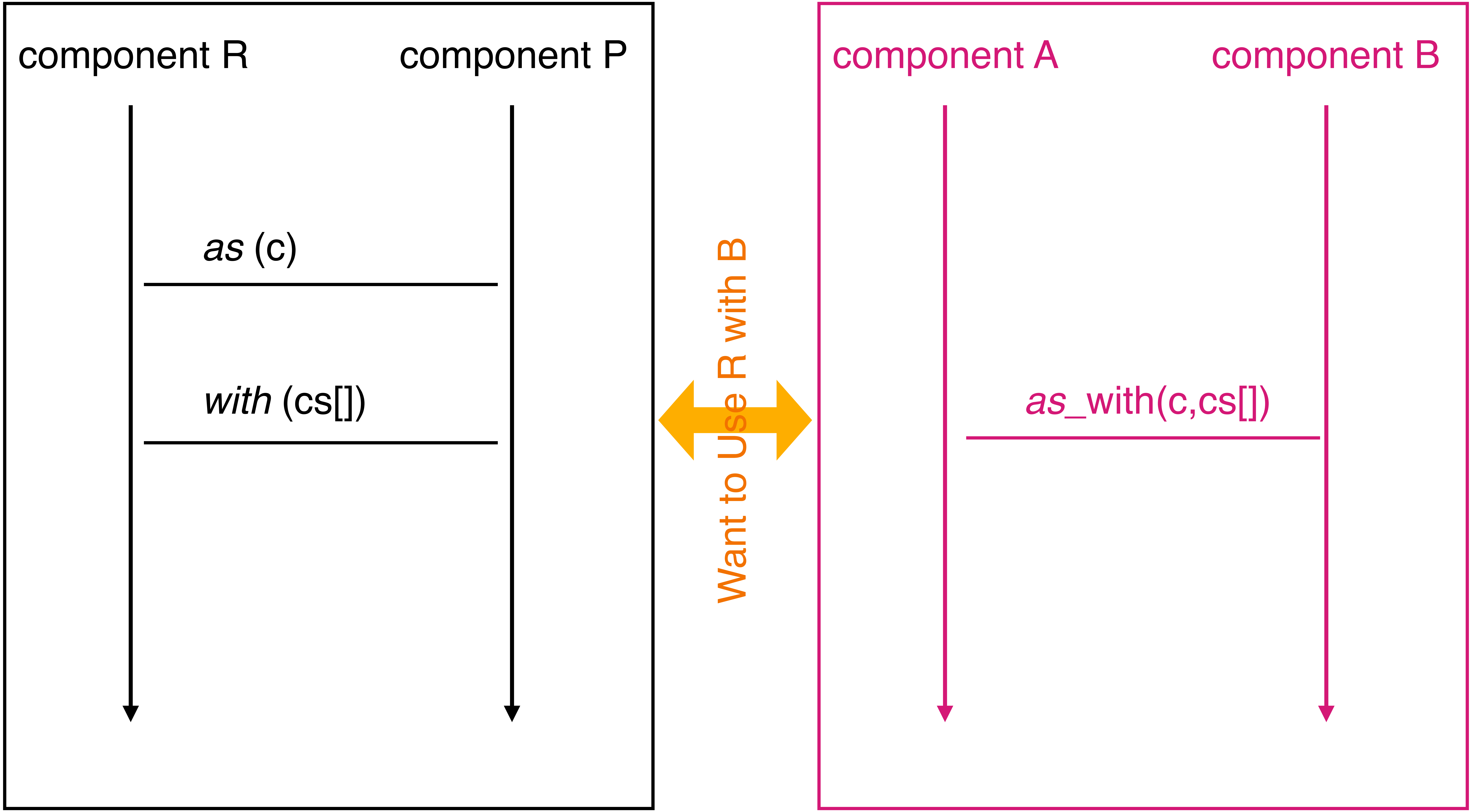

component R
sim. P for R
sim. R for P
component P
as (c)
as (c)
with (cs[])
with (cs[])

```
(define (as_is c)
  (send-on-wire (c-to-json c))

(define (with c-list)
  (send-on-wire (map c-to-json c-list)))
```

simulate P for R

```
(define (simulate-R)
  (define in (read-from-wire))
  (cond
    [(is “as” in)
     (as P (c-from-json in)))]
    [(is “with” in)
     (with P (map c-from-json in))])))

(loop-with simulate-R)
```

simulate R for P



component R

component P

as (c)

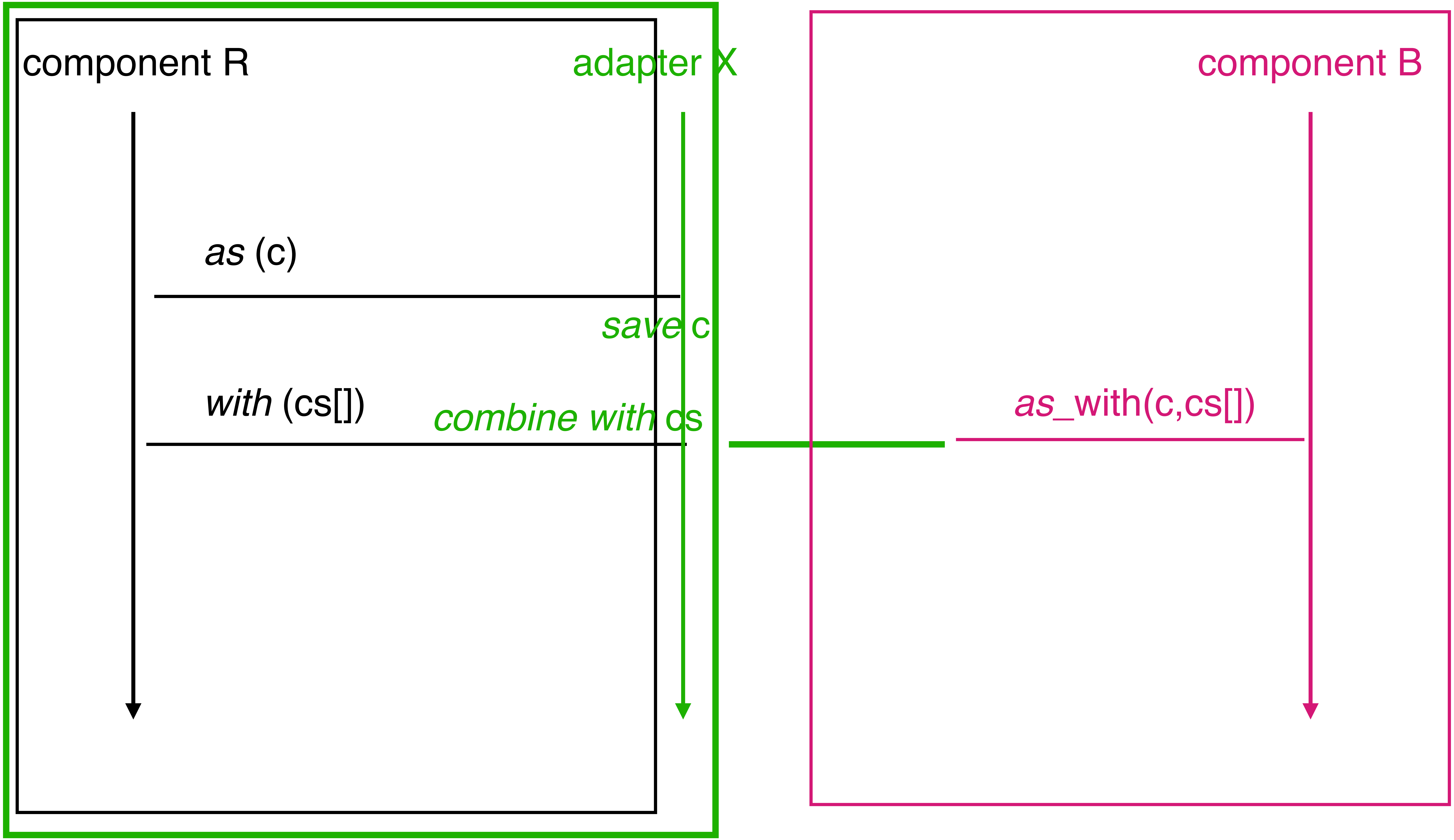
with (cs[])

Want to Use R with B

component A

component B

*as*_with(c,cs[])


component R
adapter X
component B
as (c)
save c
with (cs[])
combine with cs
*as*_with(c,cs[])