lhs and rhs passed to operator|

lhs

oChldIn

iChldOut

rhs

oChldIn

iChldOut
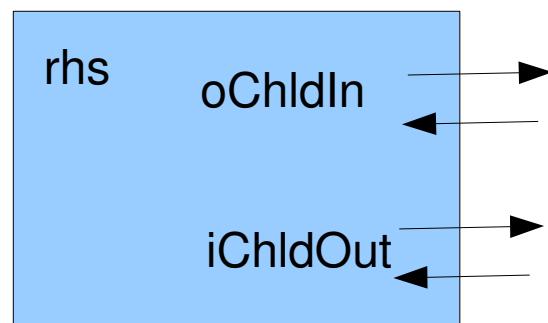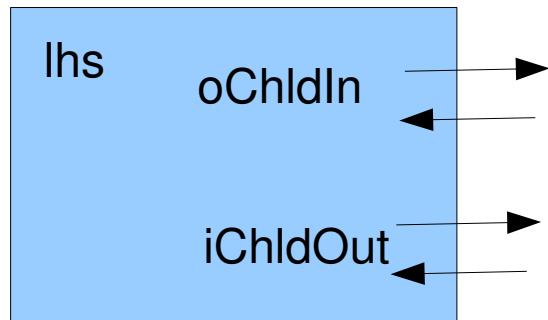
calling lhs.start(): lhs's child process starts

lhs
CE
oChldIn

lhs (child)
oChldIn

iChldOut

iChldOut

CE: CLOEXEC

rhs
oChldIn

iChldOut

P1 | P2

assigning lhs's iChildOut to rhs's oChildIn

lhs

CE
oChldIn

iChldOut

CE: CLOEXEC

lhs
(child)   oChldIn

iChldOut

rhs   oChldIn

iChldOut
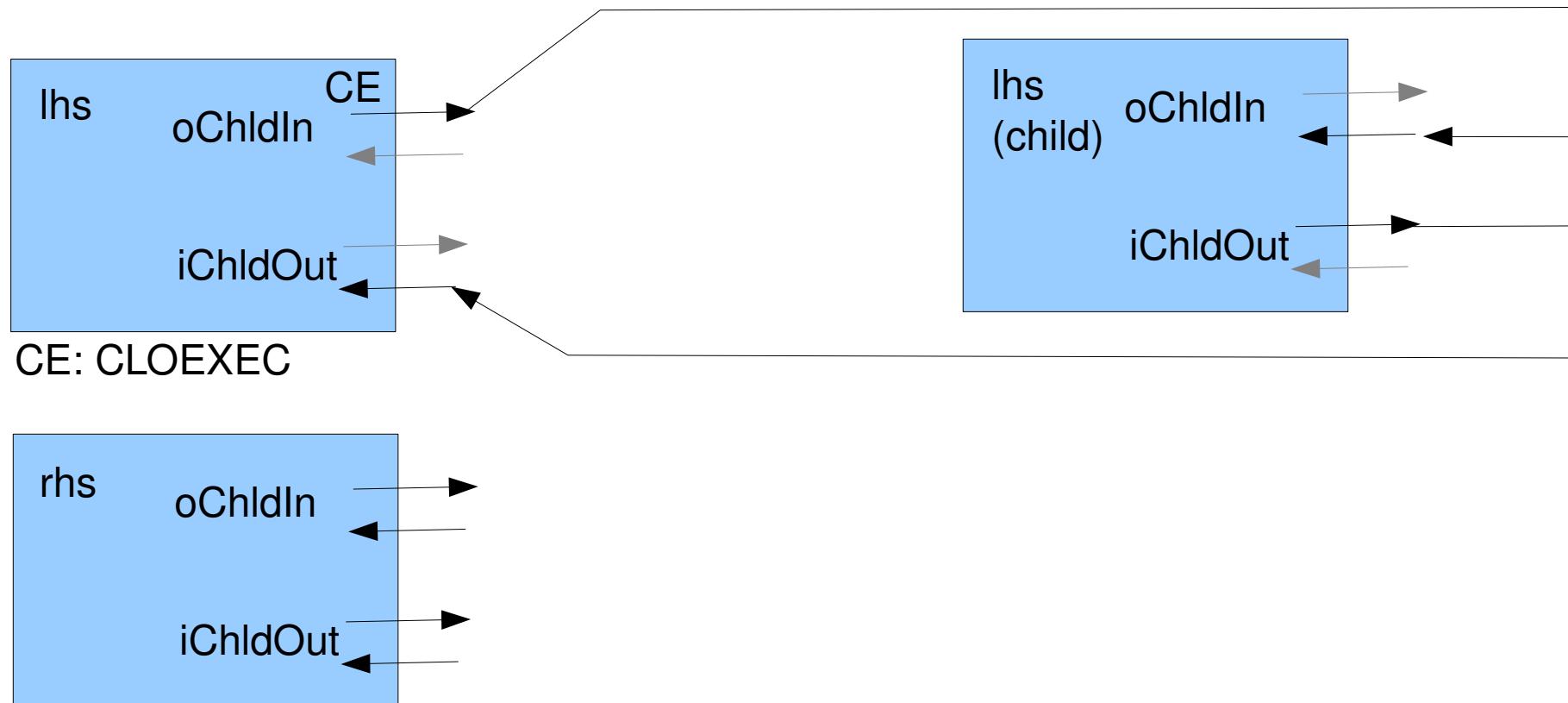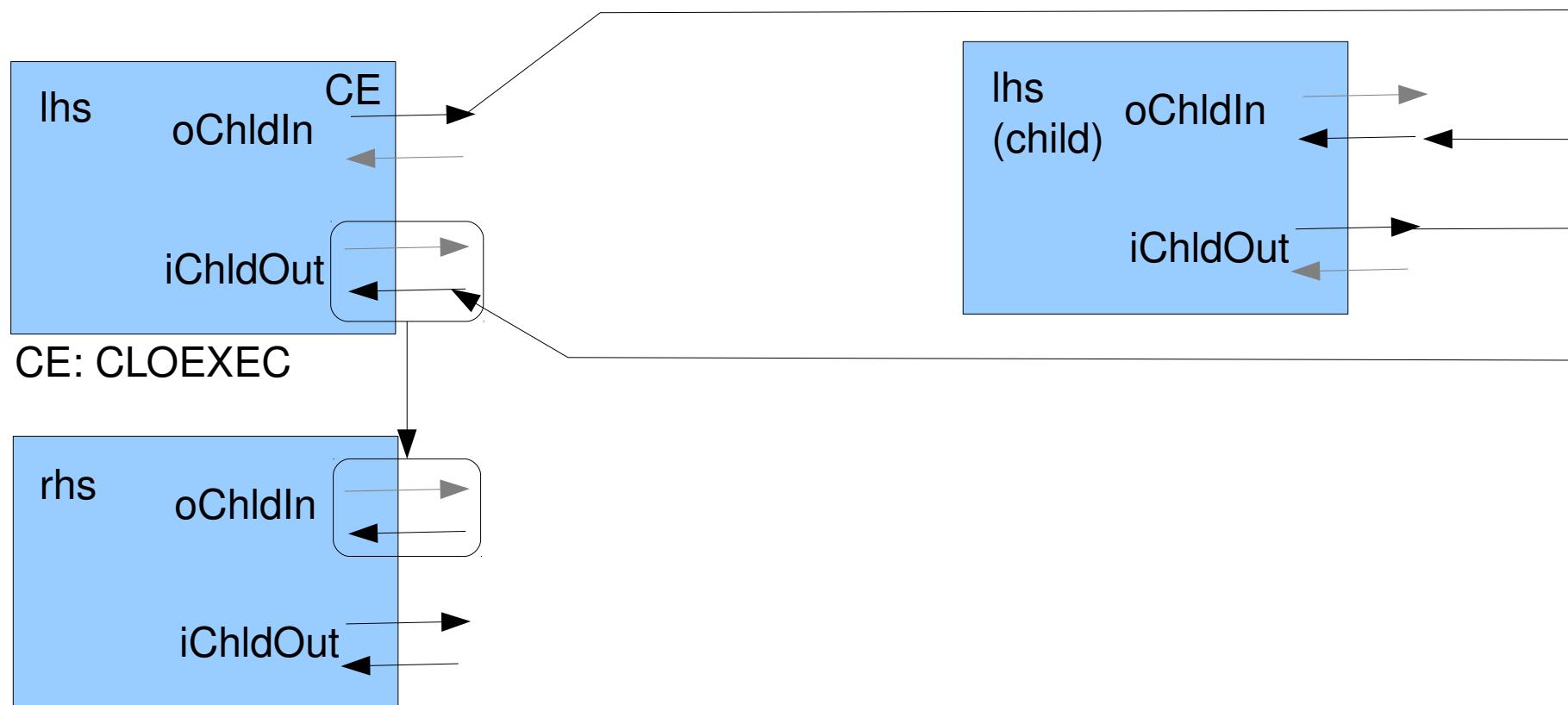
P1 | P2 | P3   passing P2 (lhs) and P3 (rhs) to operator|

P1

CE
oChldIn

iChldOut

CE: CLOEXEC

P1 (child)   oChldIn

iChldOut

lhs

oChldIn

iChldOut

rhs

oChldIn

iChldOut
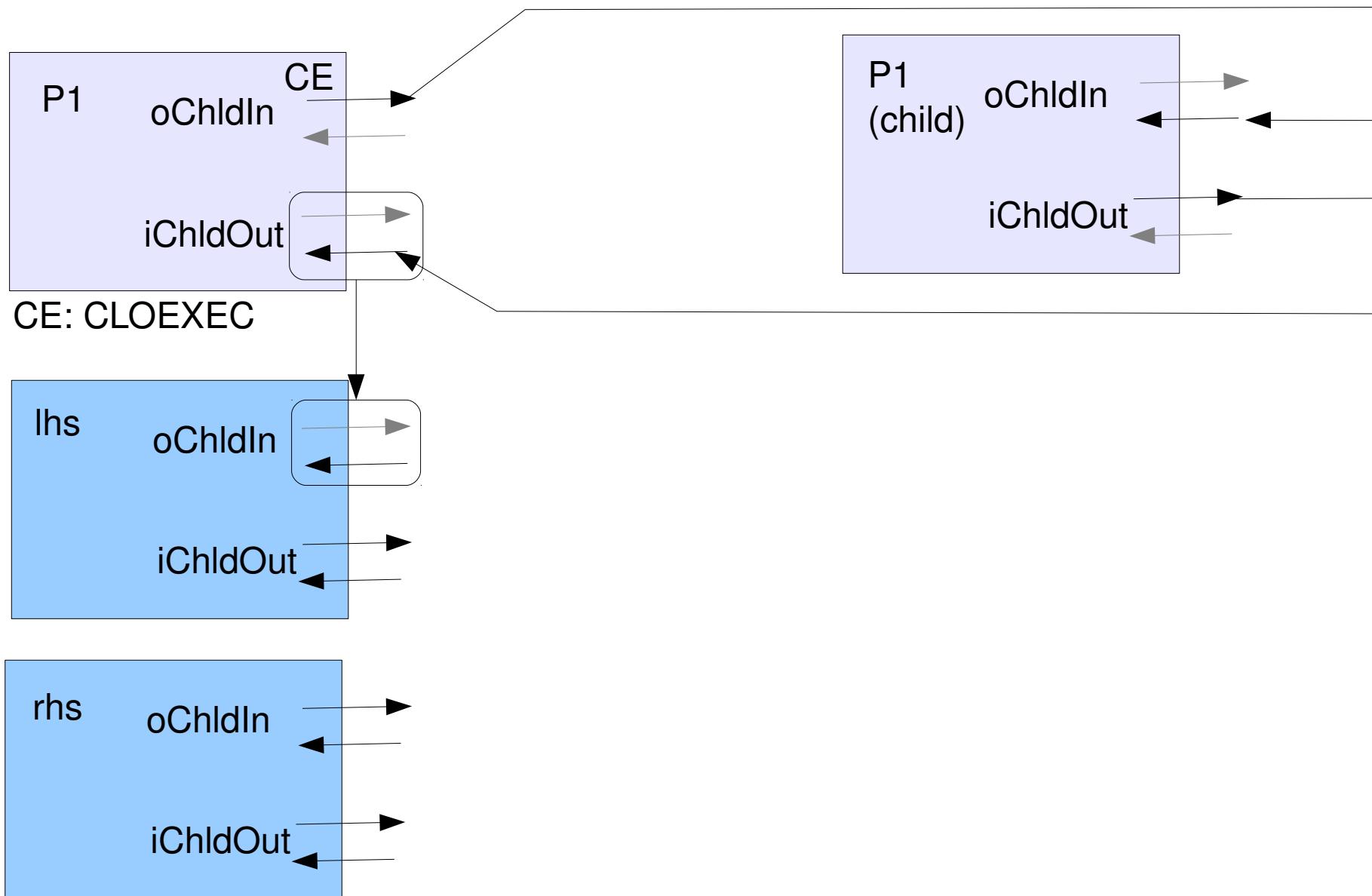
P1 | P2 | P3   calling lhs.start()

P1
oChldIn   CE

iChldOut

CE: CLOEXEC

lhs
oChldIn

iChldOut

rhs
oChldIn

iChldOut

P1
(child)   oChldIn

iChldOut

CE   (closes)
P1'soChldIn

lhs
(child)   oChldIn

iChldOut
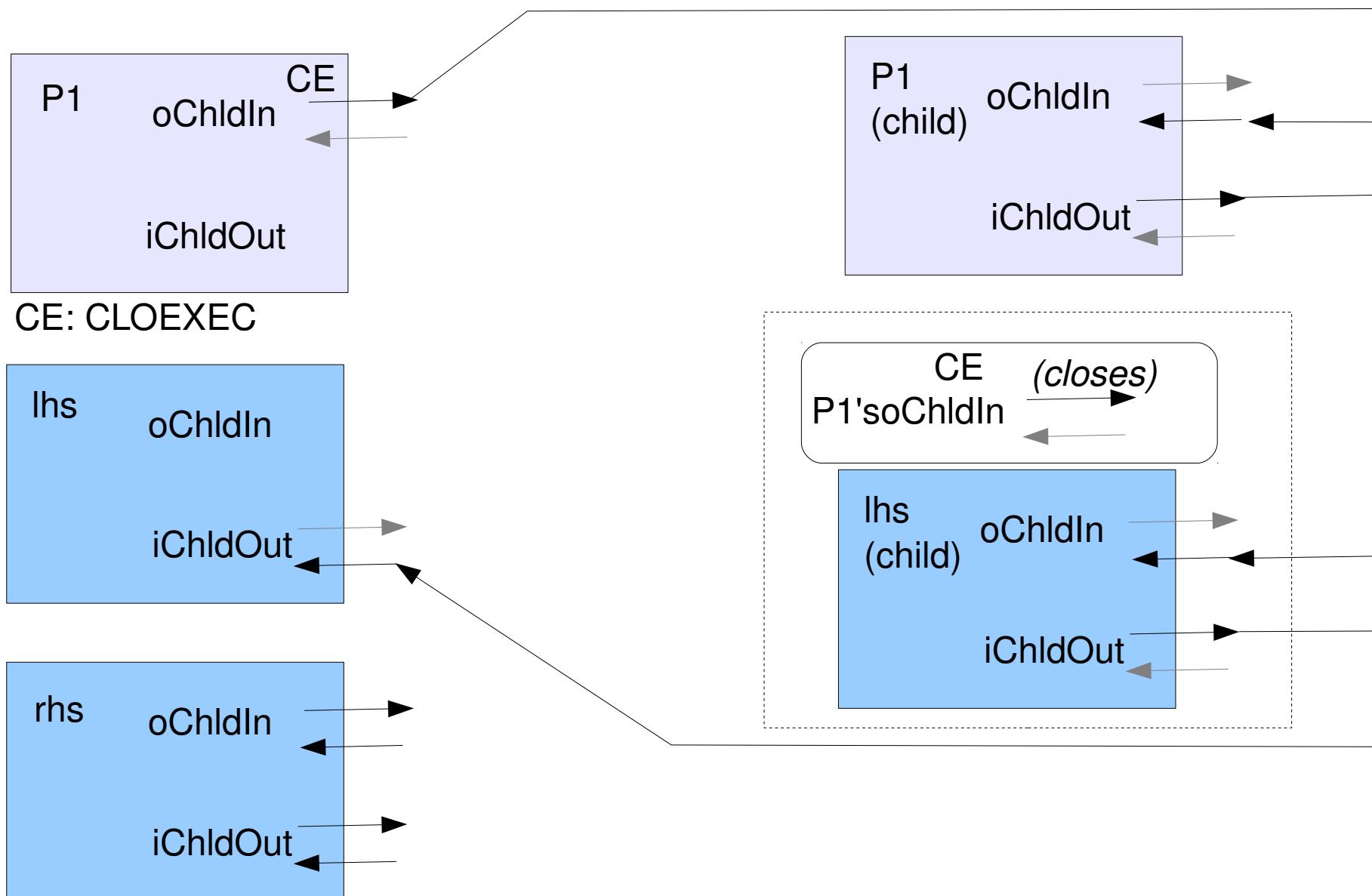
P1 | P2 | P3   assigning lhs's iChildOut to rhs's oChildIn

P1
oChldIn   CE

iChldOut

CE: CLOEXEC

lhs   oChldIn

iChldOut

rhs   oChldIn

iChldOut

P1
(child)   oChldIn

iChldOut

CE   *(closes)*
P1'soChldIn

lhs
(child)   oChldIn

iChldOut
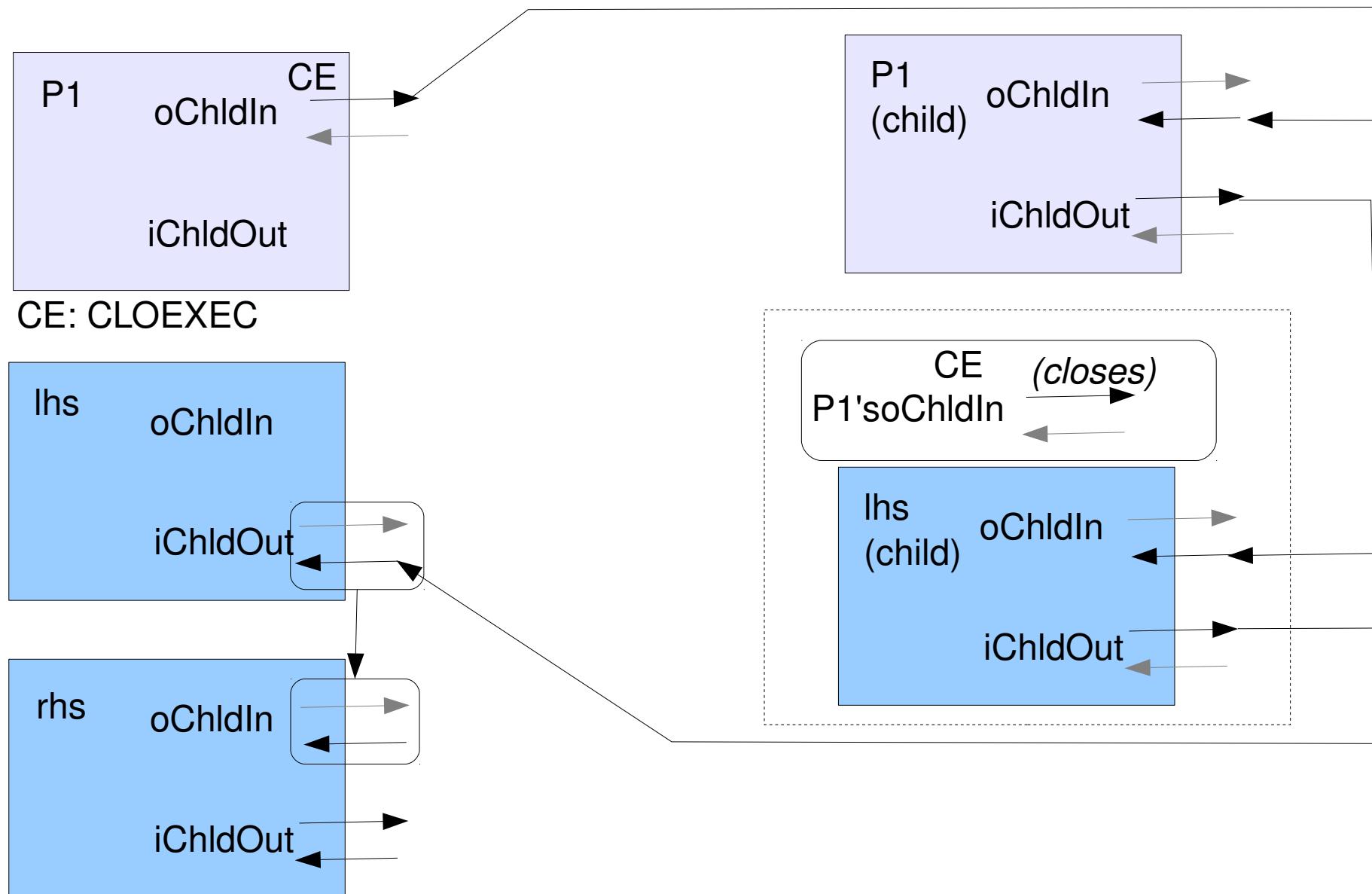
(P1 | P2 | P3).start()

calling rhs.start()

P1

CE

oChldIn

iChldOut

CE: CLOEXEC

lhs

oChldIn

iChldOut

rhs

oChldIn

iChldOut

P1 (child)

oChldIn

iChldOut

lhs (child)

oChldIn

iChldOut

CE
P1'soChldIn

(closes)

lhs (child)

oChldIn

iChldOut