SummerSoc 2026 Χερσονισος, Ελλαδα

Trust by design – in praise of modularization: a case study

Wolfgang Reisig

Humboldt-Universität zu Berlin Germany

Peter Fettke

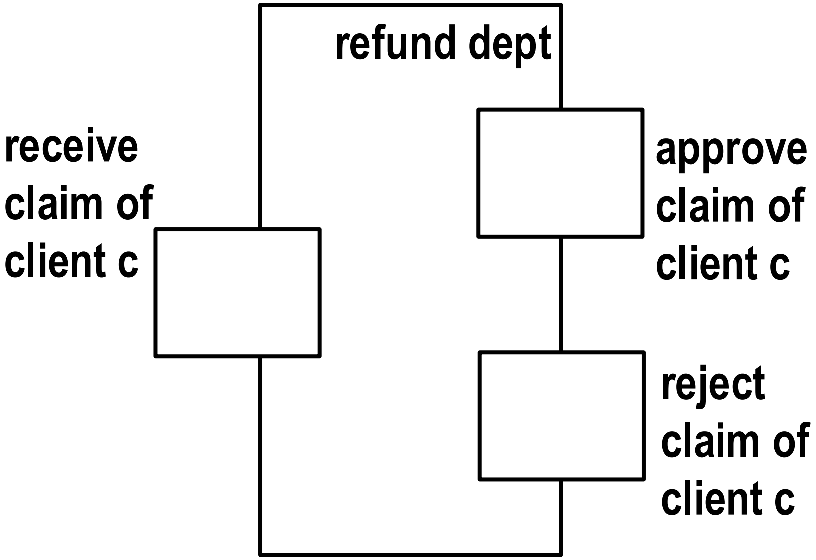
German Research Center for Artificial Intelligence (DFKI) and Saarland University, Saarbrücken, Germany

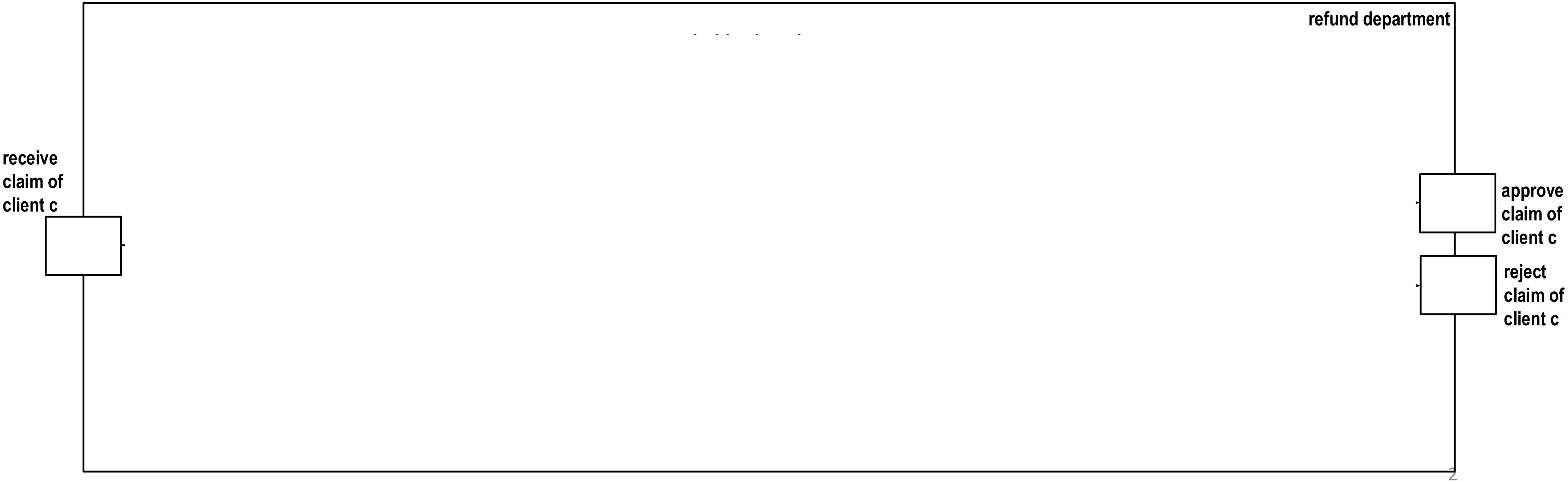
Institut für Informatik
Humboldt-Universität zu Berlin

Deutsches Forschungszentrum für Künstliche Intelligenz
German Research Center for Artificial Intelligence

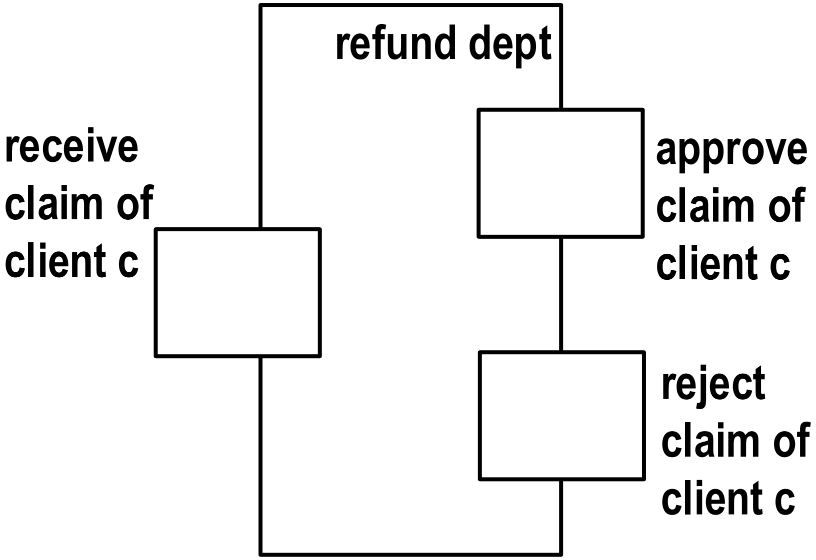
UNIVERSITÄT DES SAARLANDES

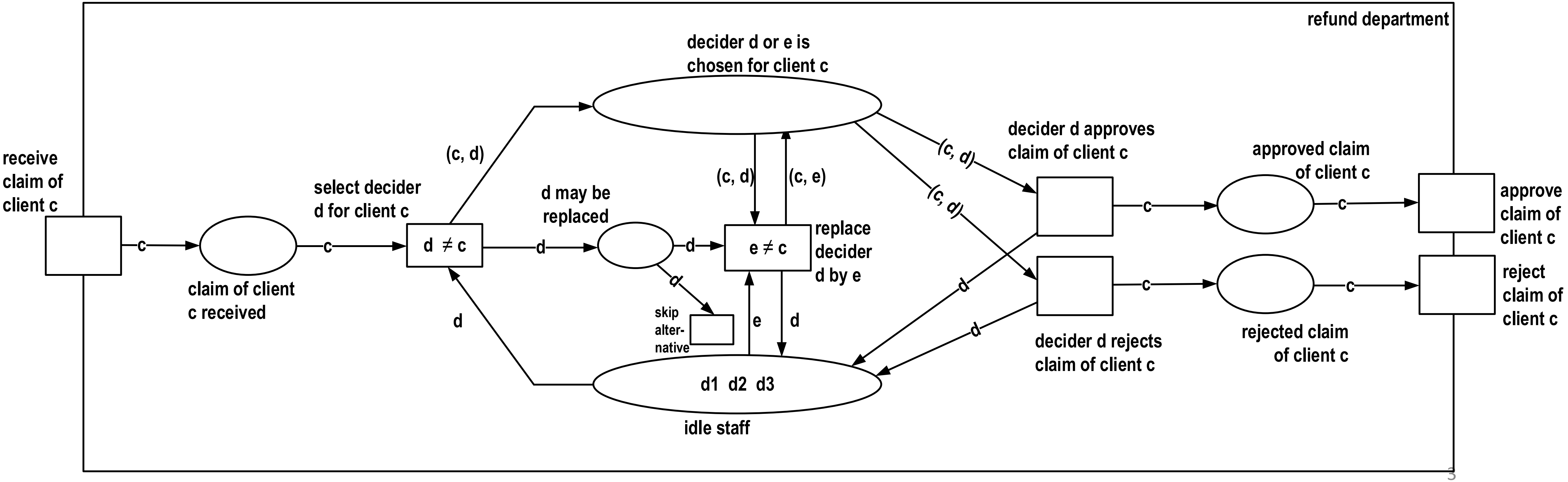
most abstract view



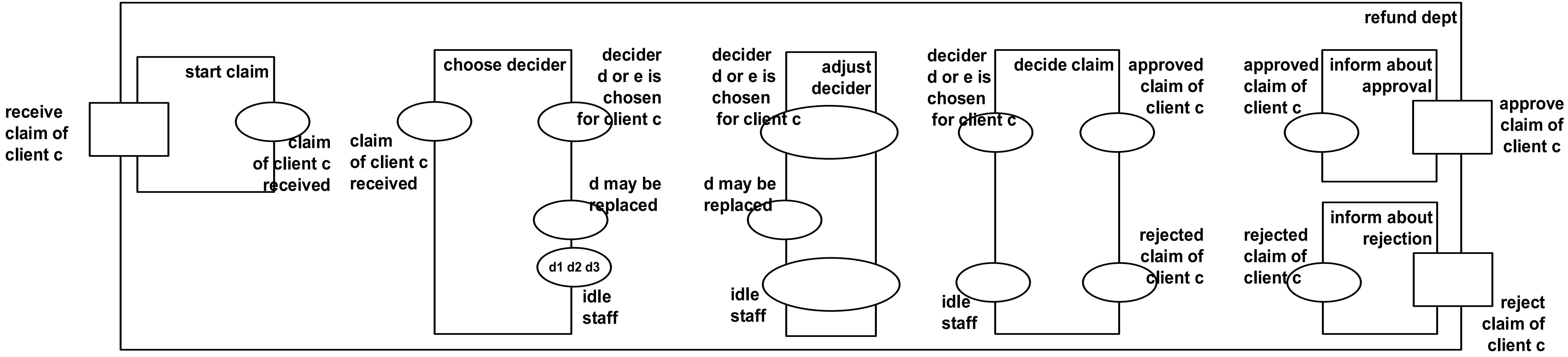


… grasp this in one go?





better: six modules of the system model

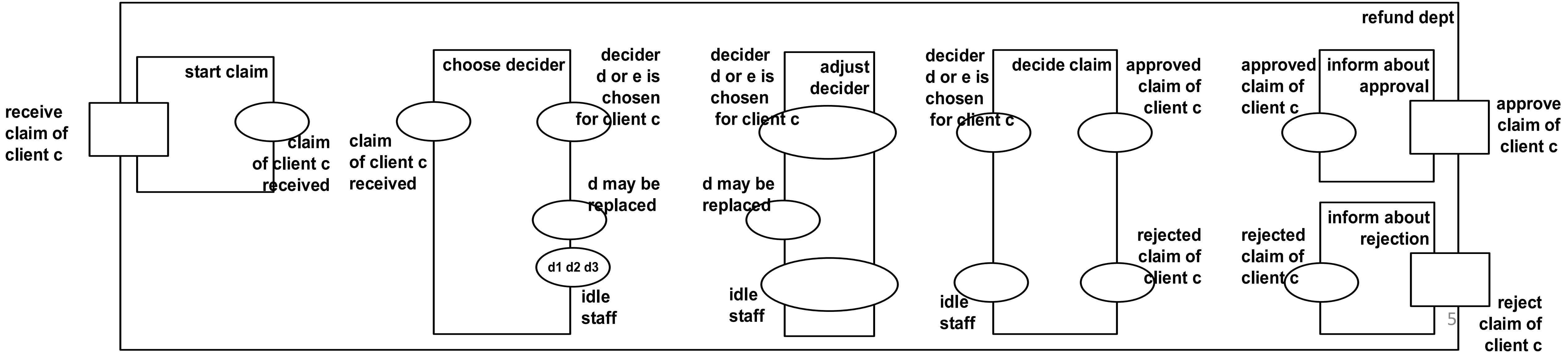


This talk

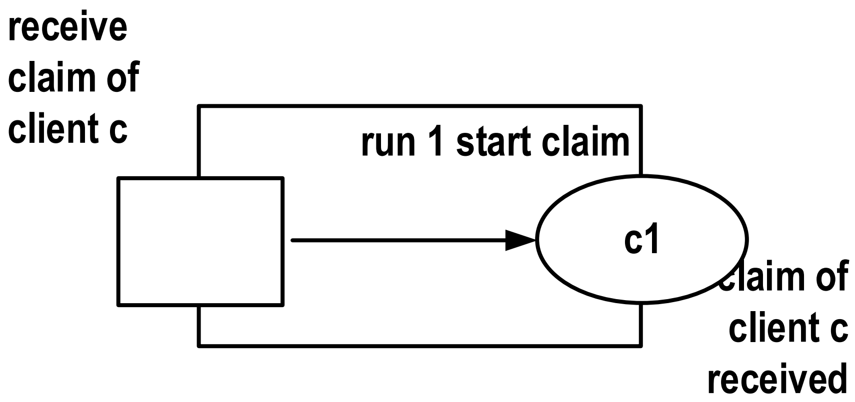
Part I Run modules and typical runs

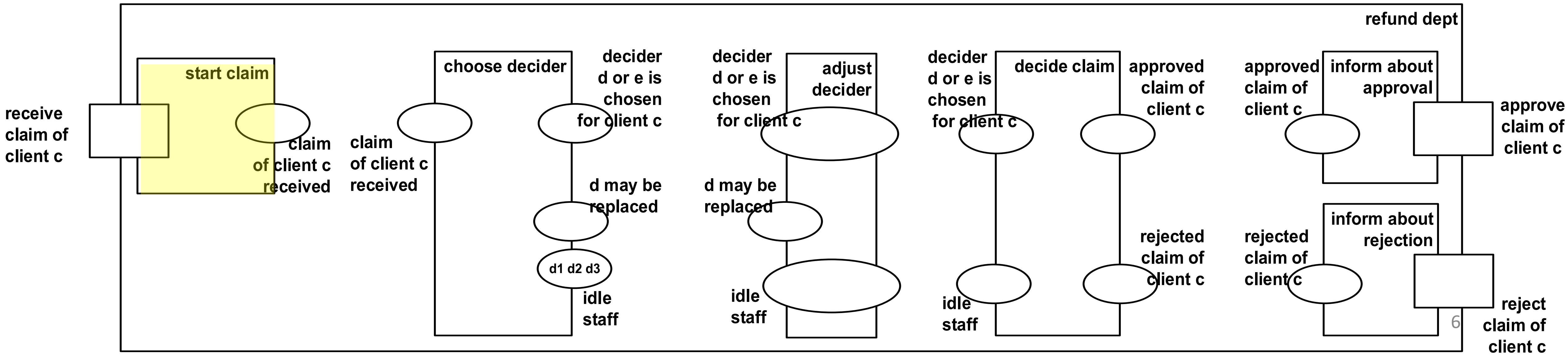
Part II System modules

Part III Liveness

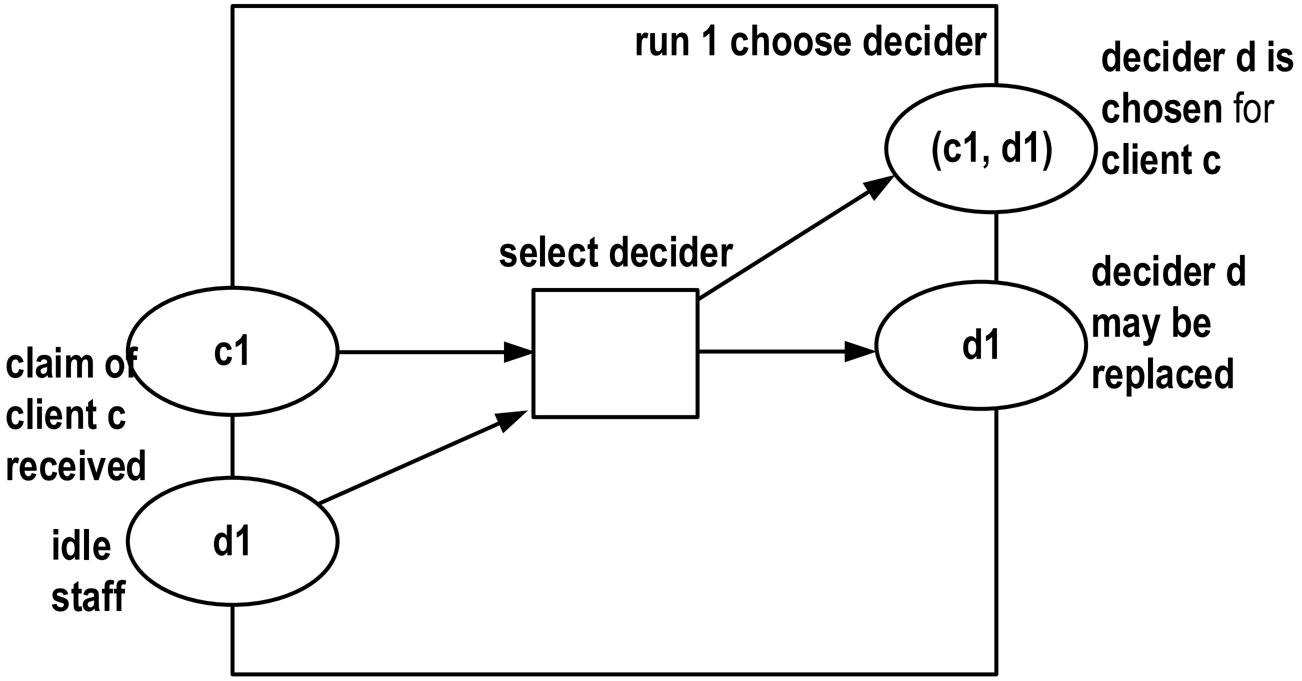


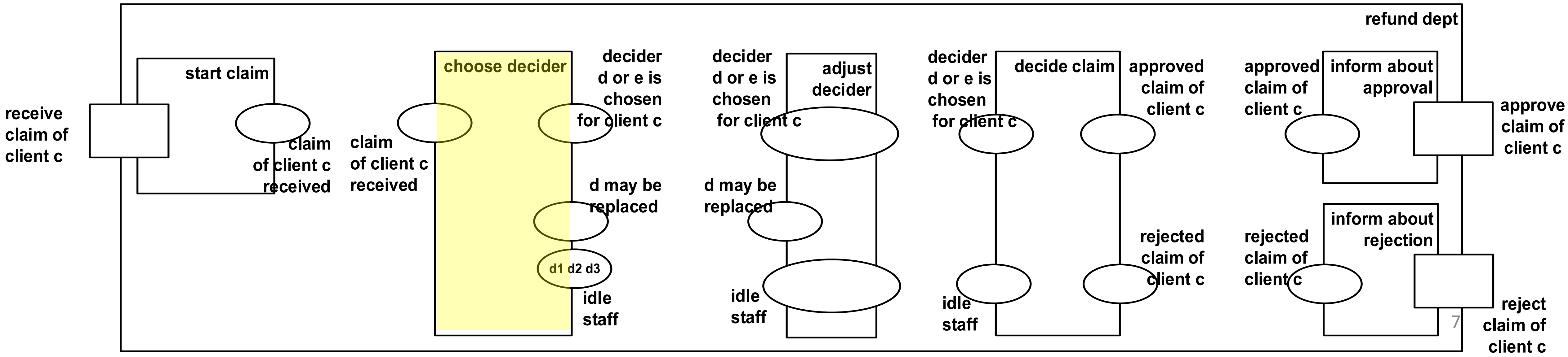
Part I run modules



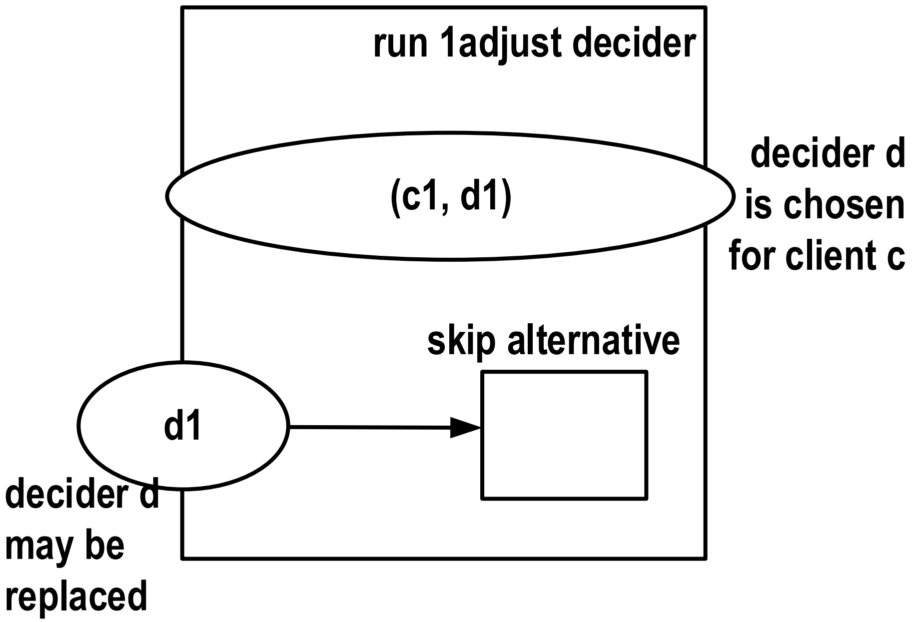


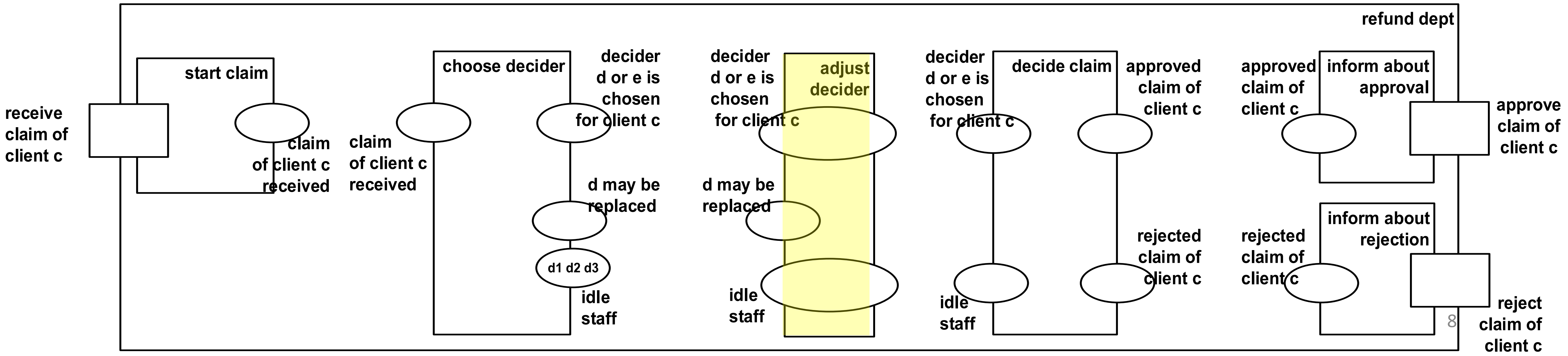
Part I run modules



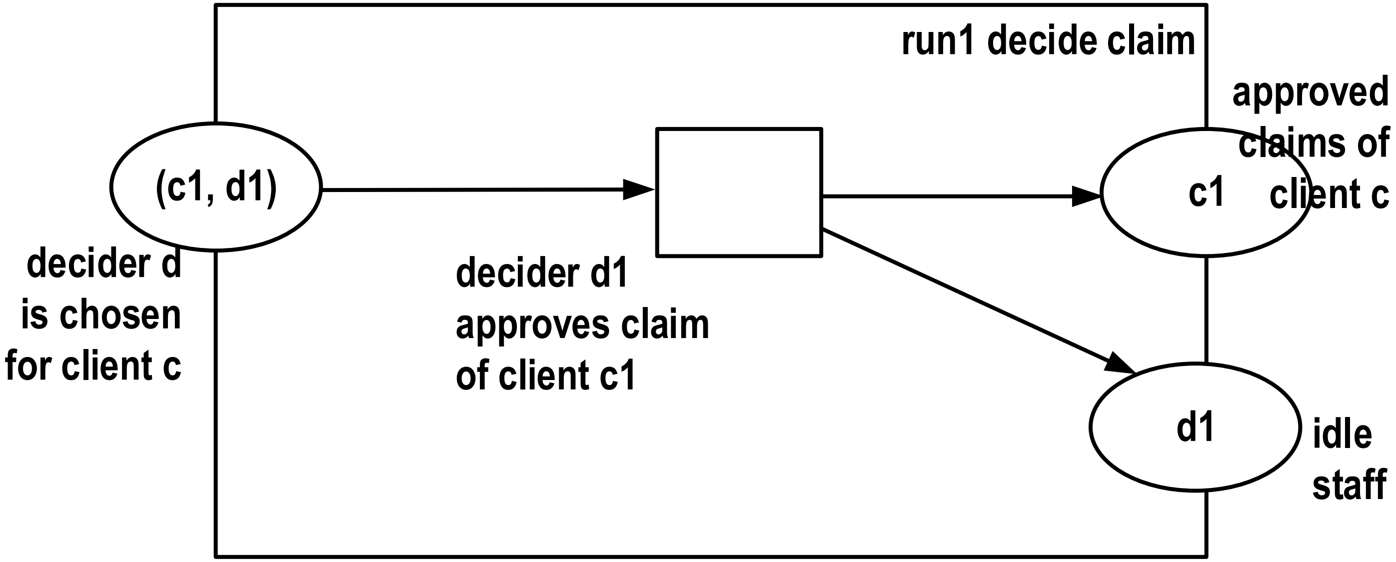


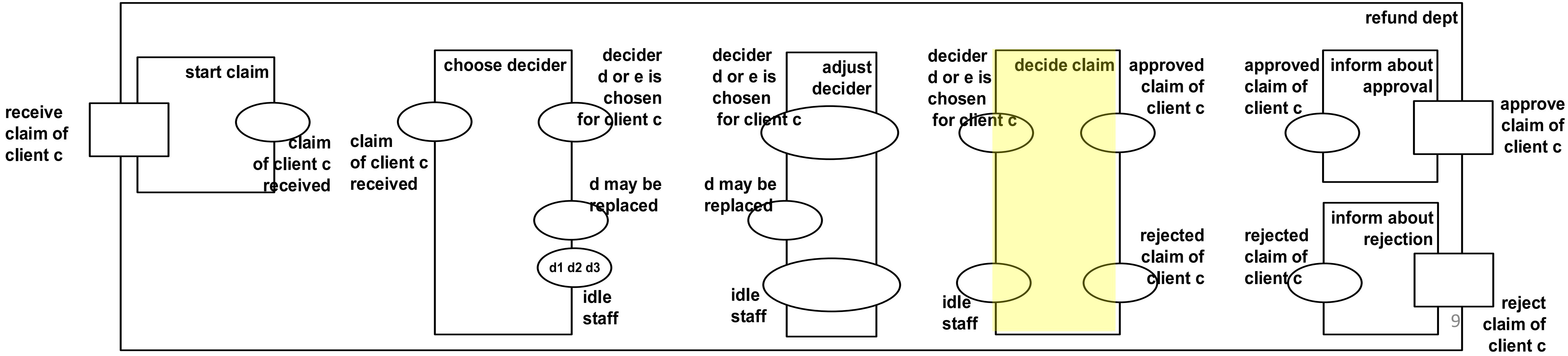
Part I run modules



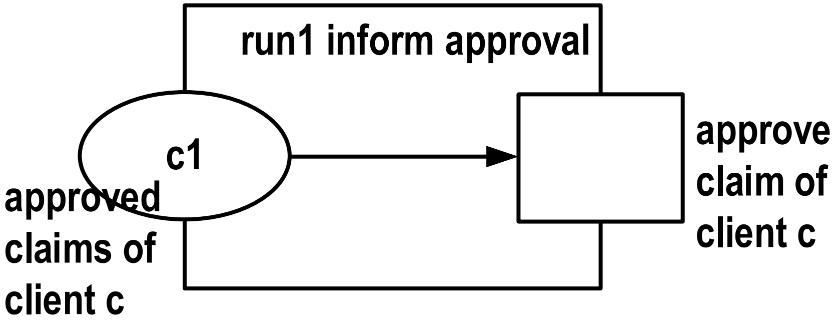


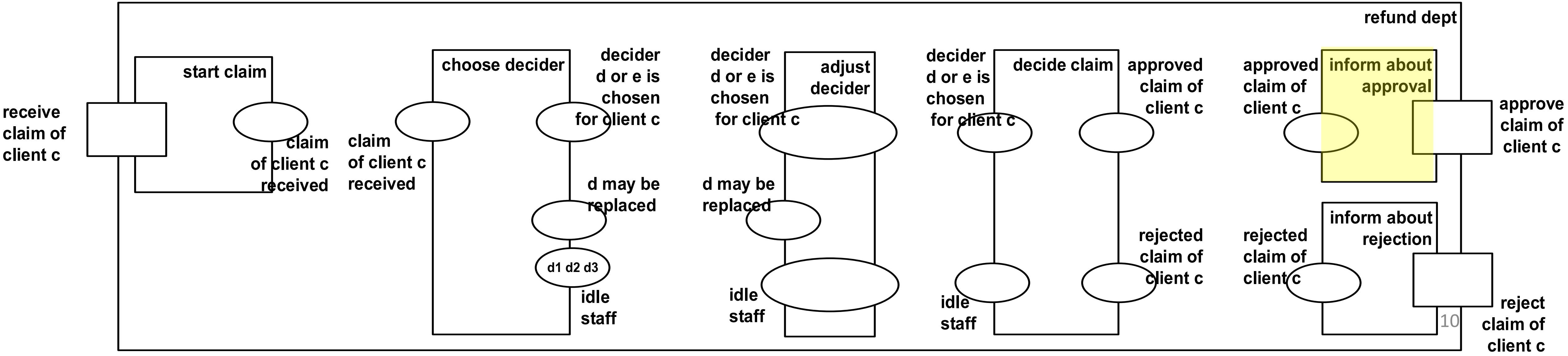
Part I run modules

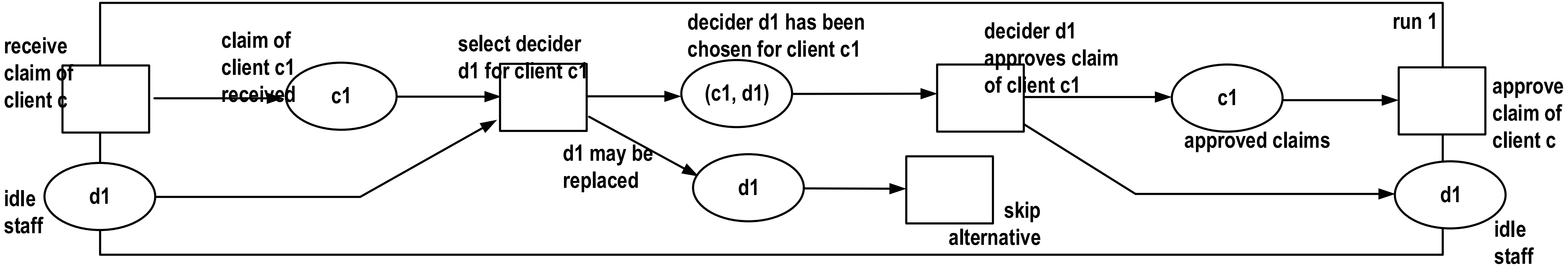




Part I run modules

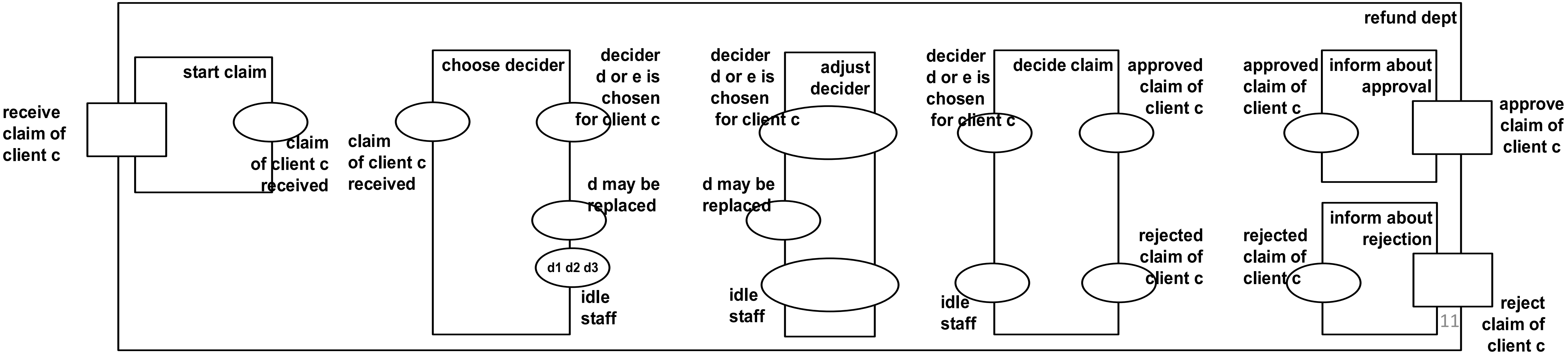


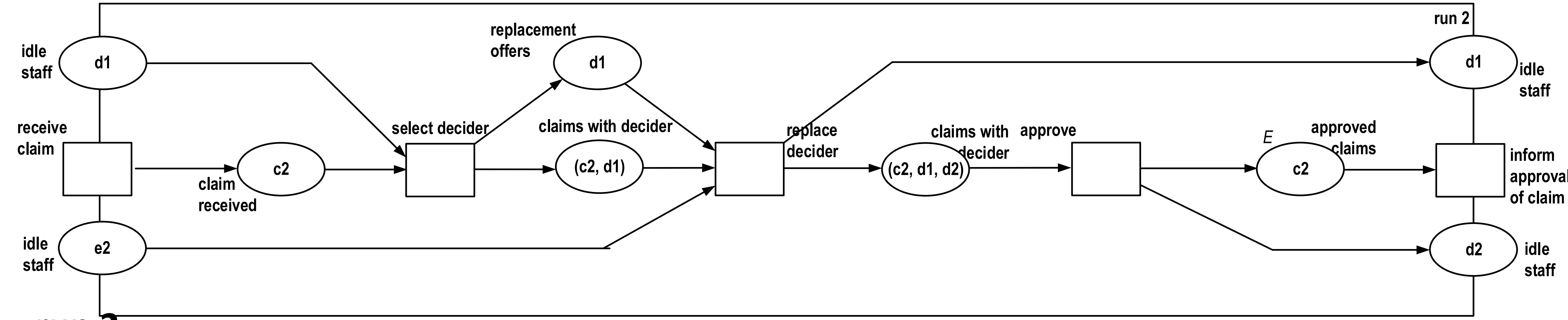




run1 =

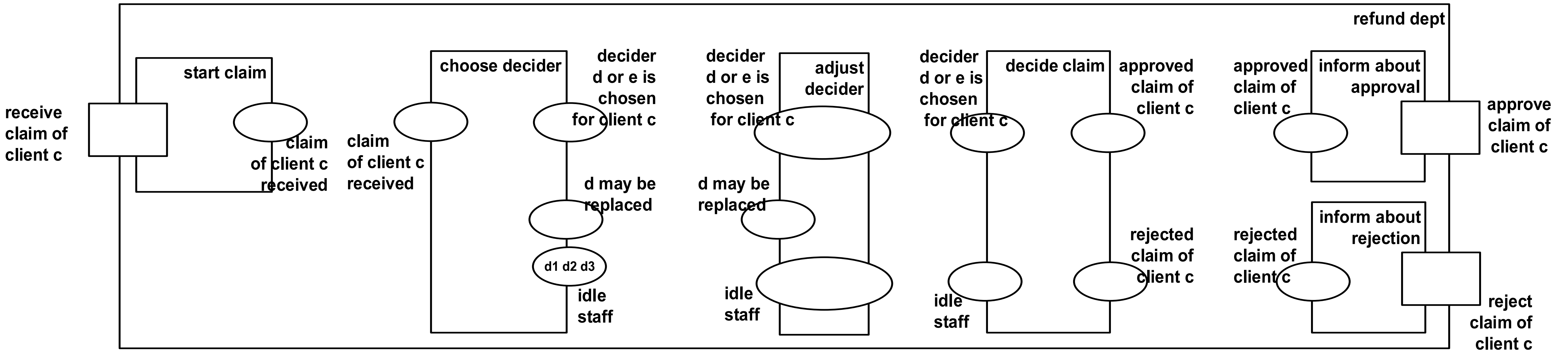
run 1 start claim • run1 choose decider • run1 adjust decider • run1 decide claim • run1 inform about approval



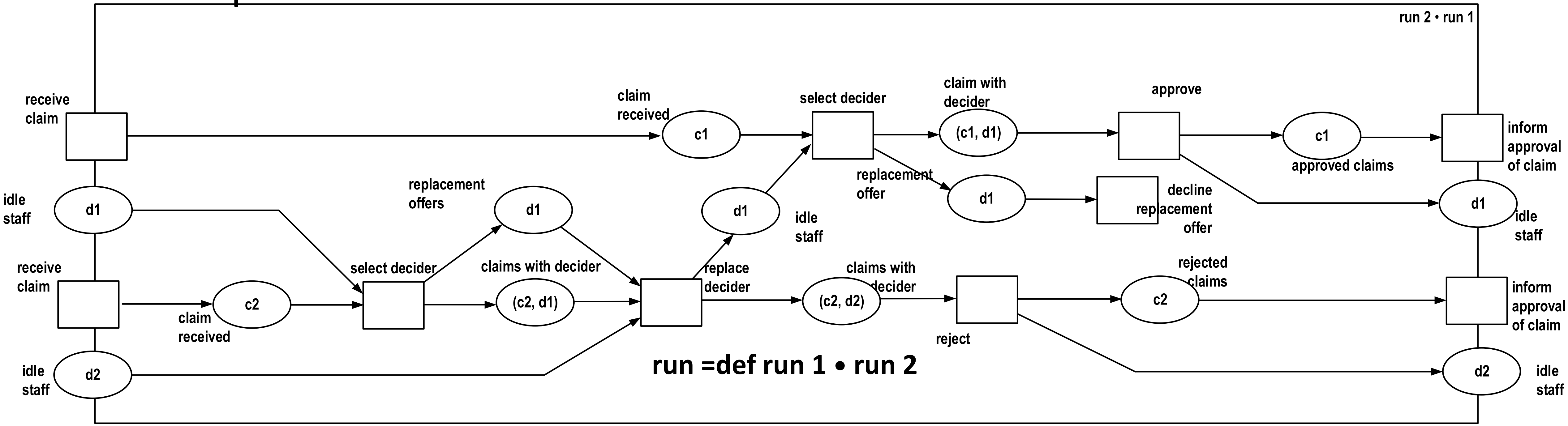


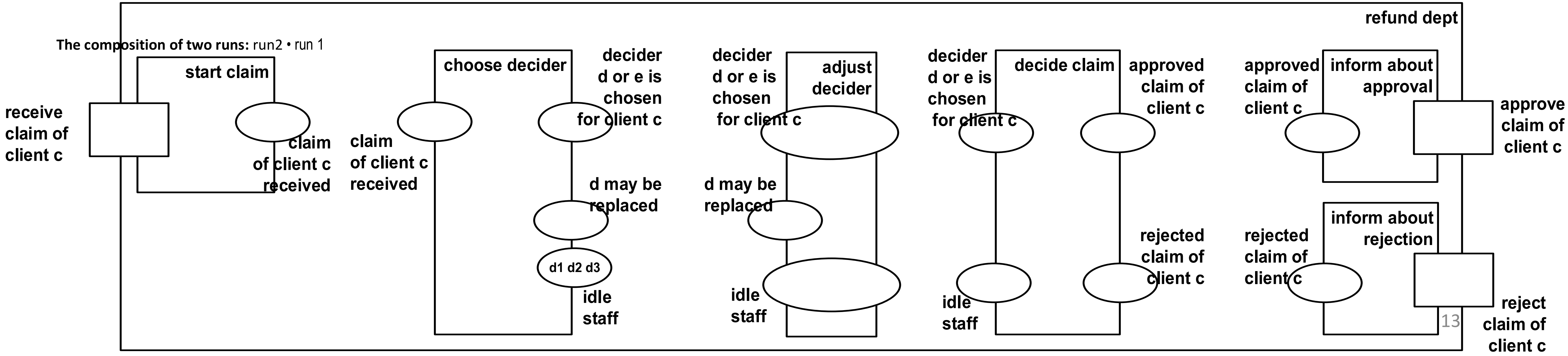
run 2 =

run 2 start claim • run2 choose decider • run2 adjust decider • run2 decide claim • run2 inform about rejection

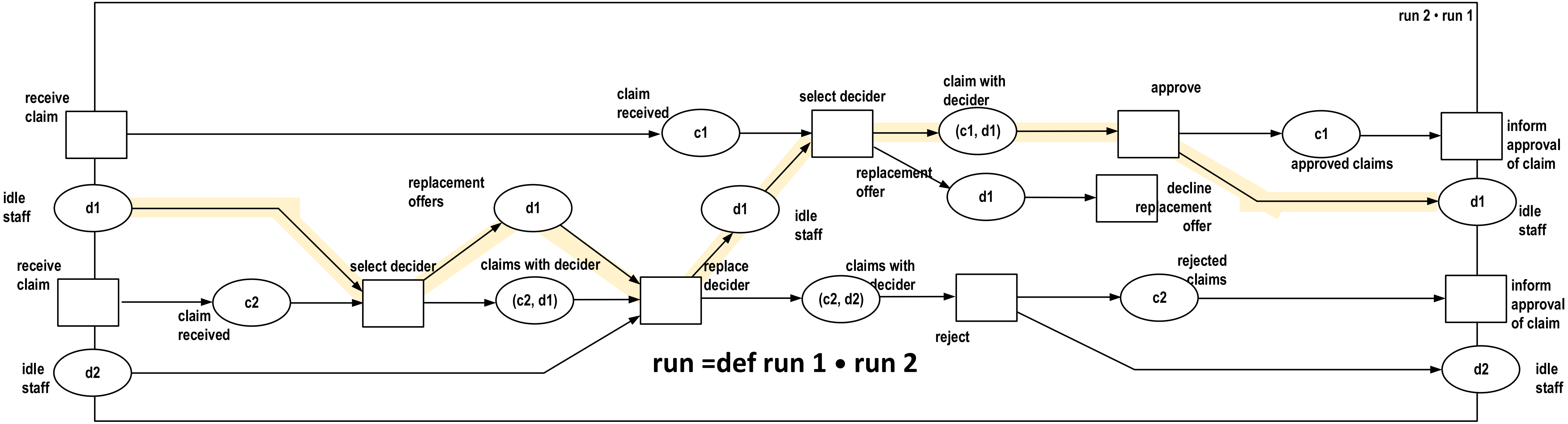


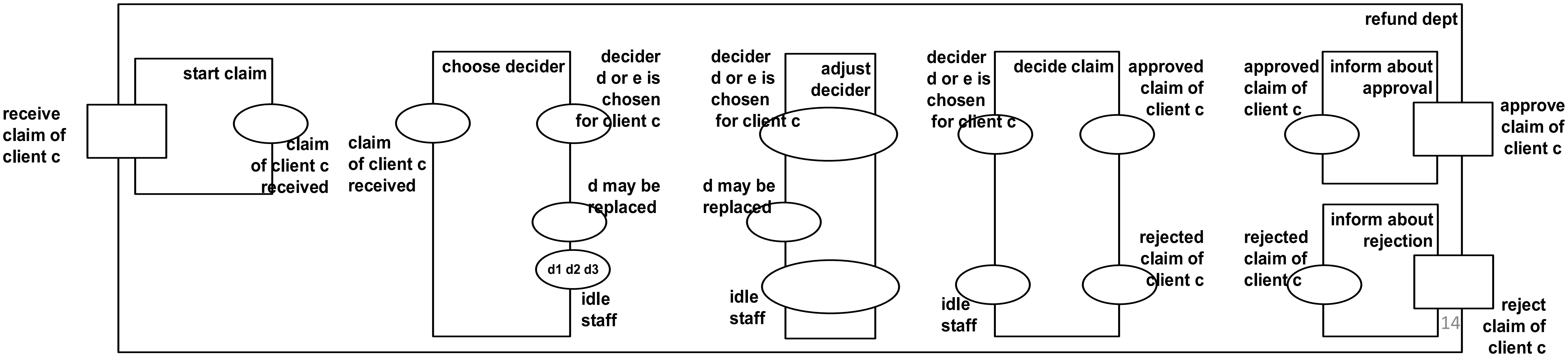
composition of both runs



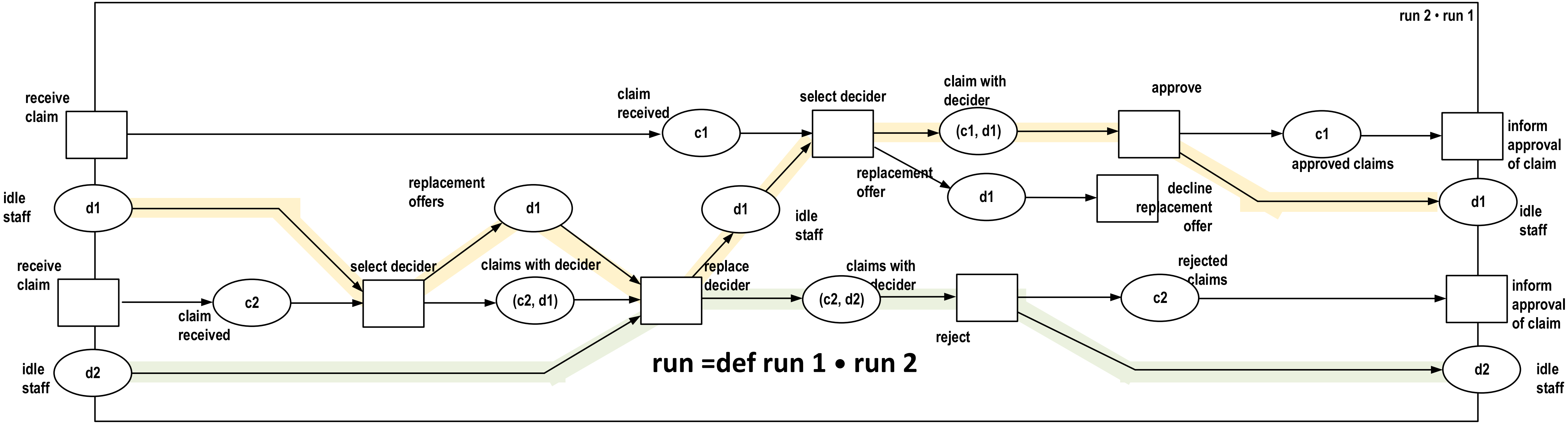


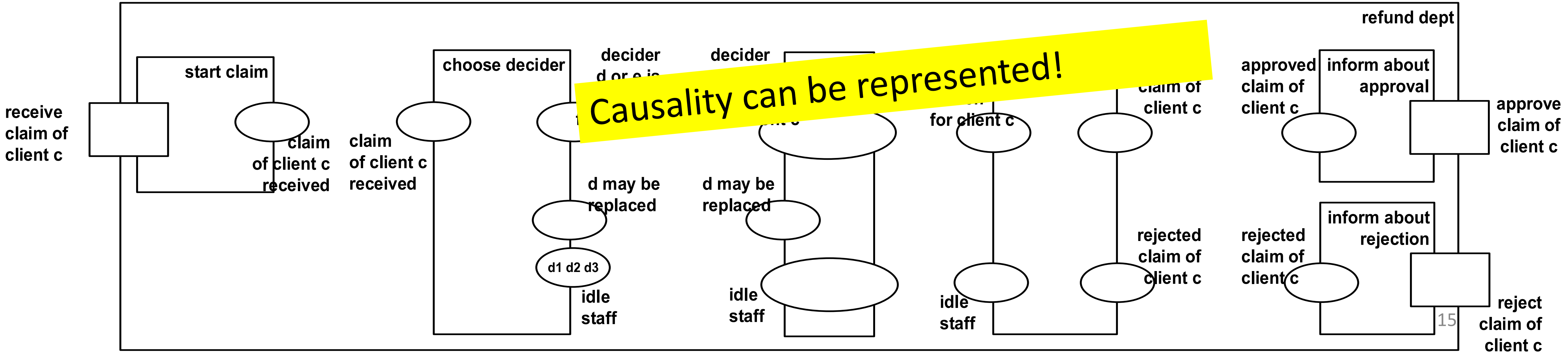
the lifeline of d1





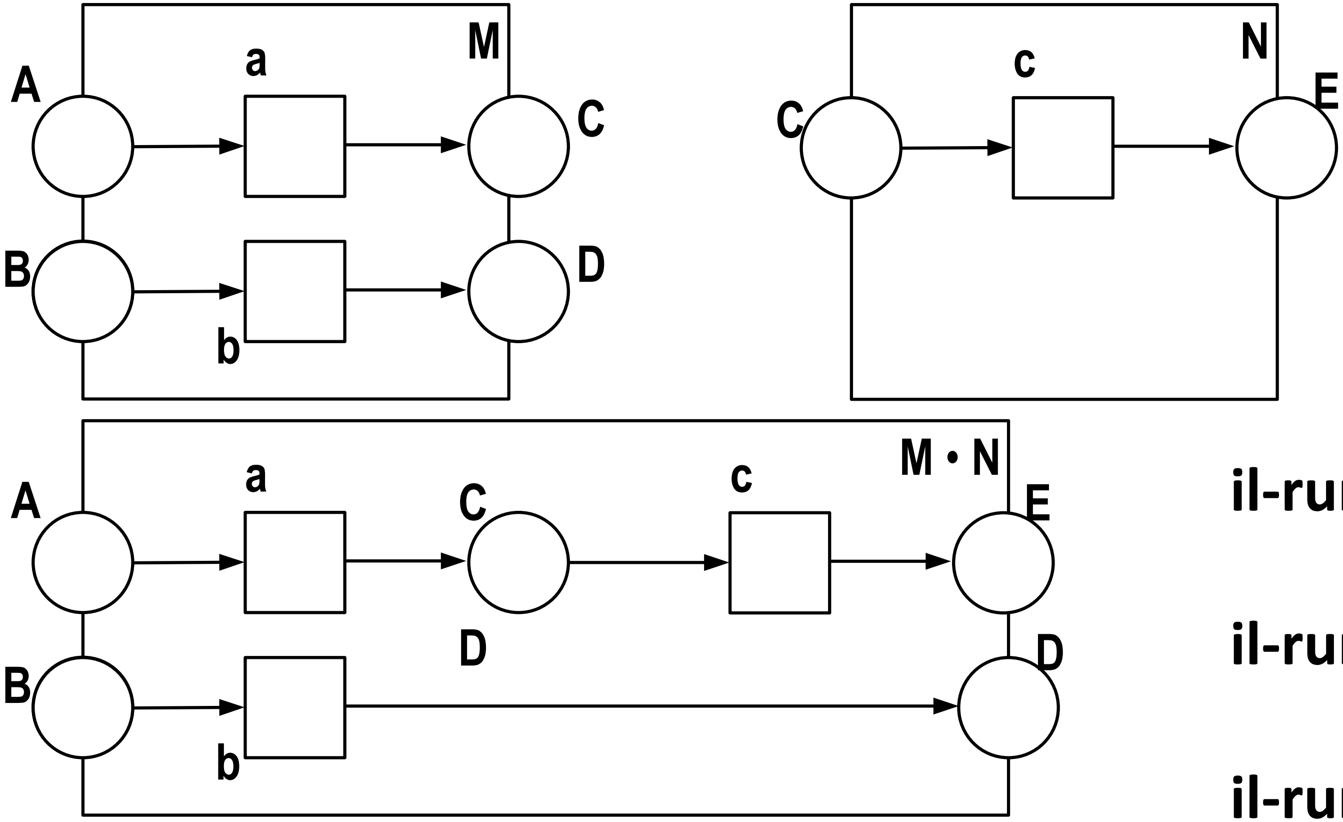
the lifeline of d1 and of d2





Causality can be represented!

runs are compositional!



interleaved runs are not compositional!

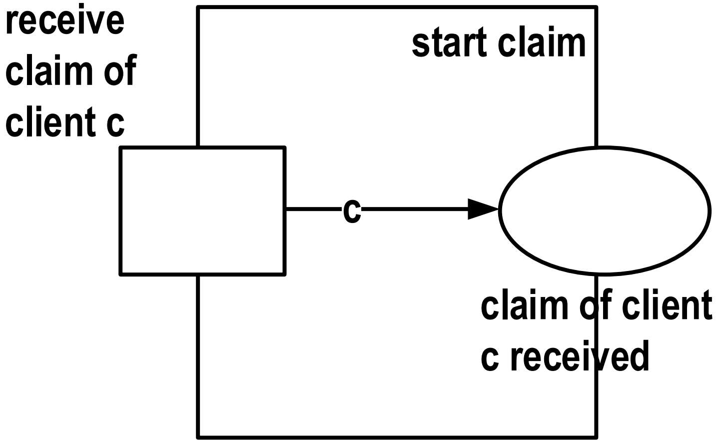
il-runs(M • N) = {a-b-c, b-a-c, a-c-b}

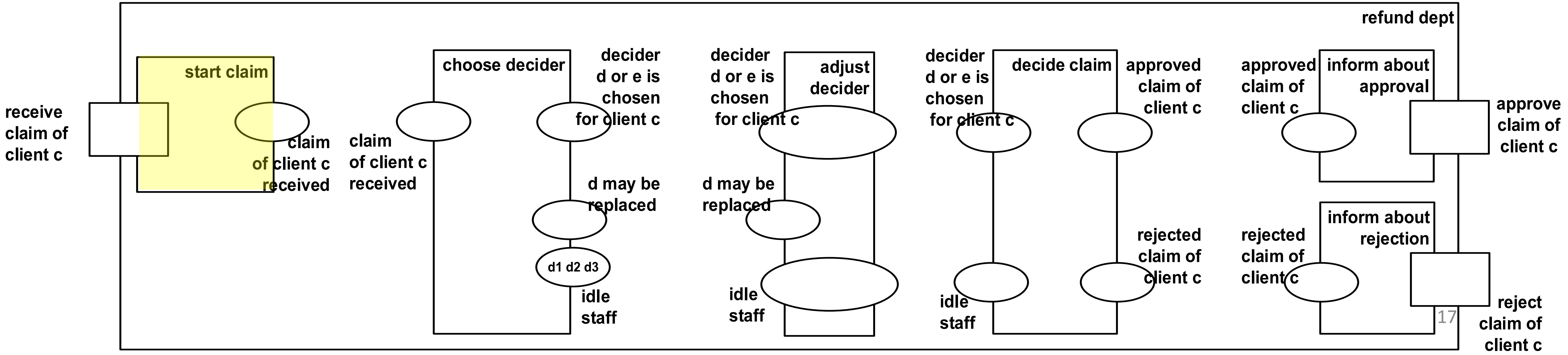
il-runs(M) = {a-b, b-a} il-runs runs(N) = {c}

il-runs(M) • il-runs runs(N) = {a-b-c, b-a-c}

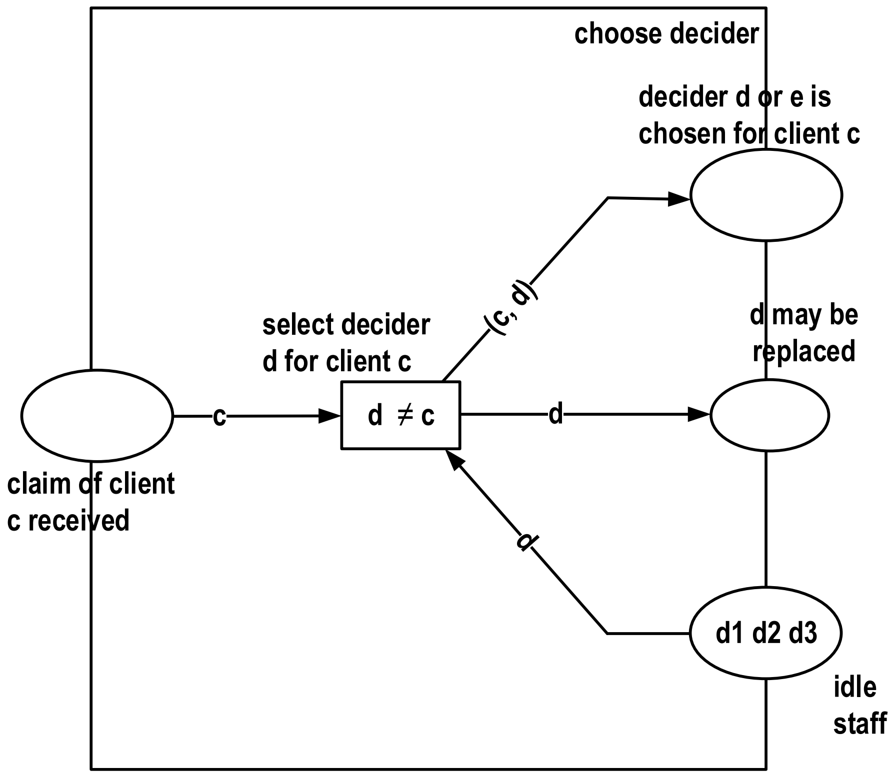
runs(M • N) = runs(M) • runs(N)

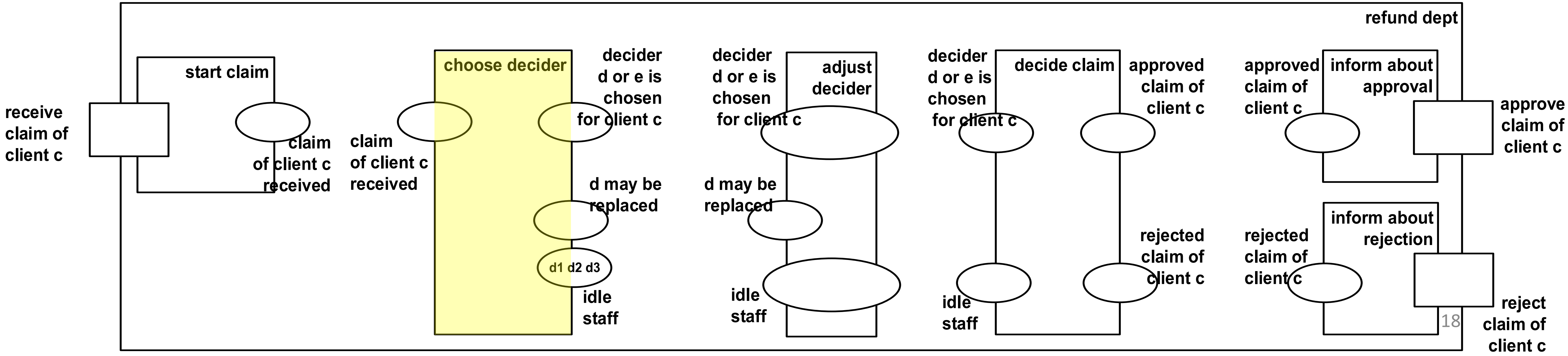
Part II system modules



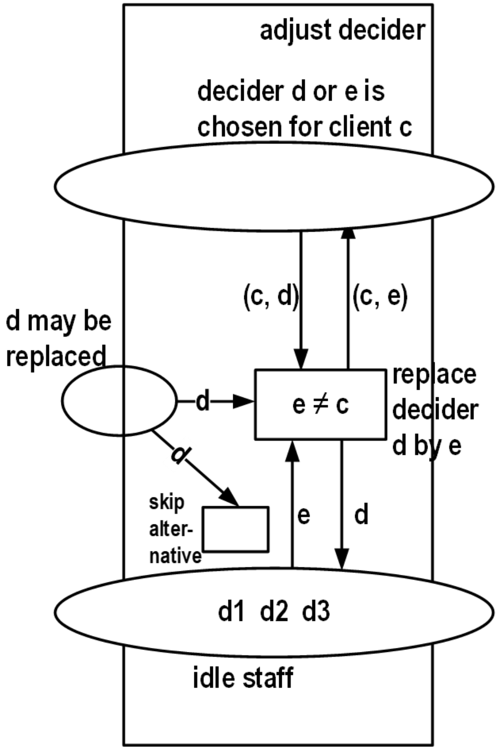


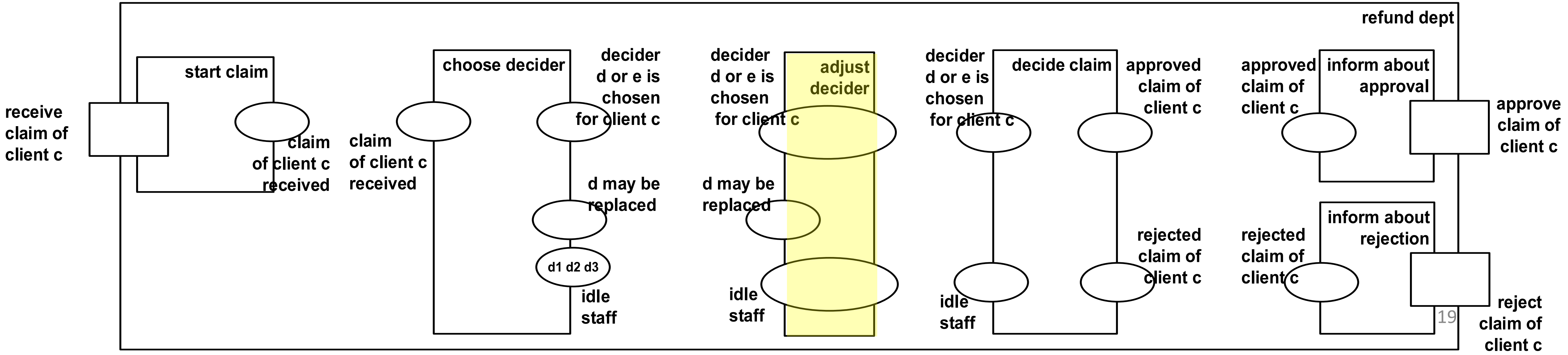
Part II system modules

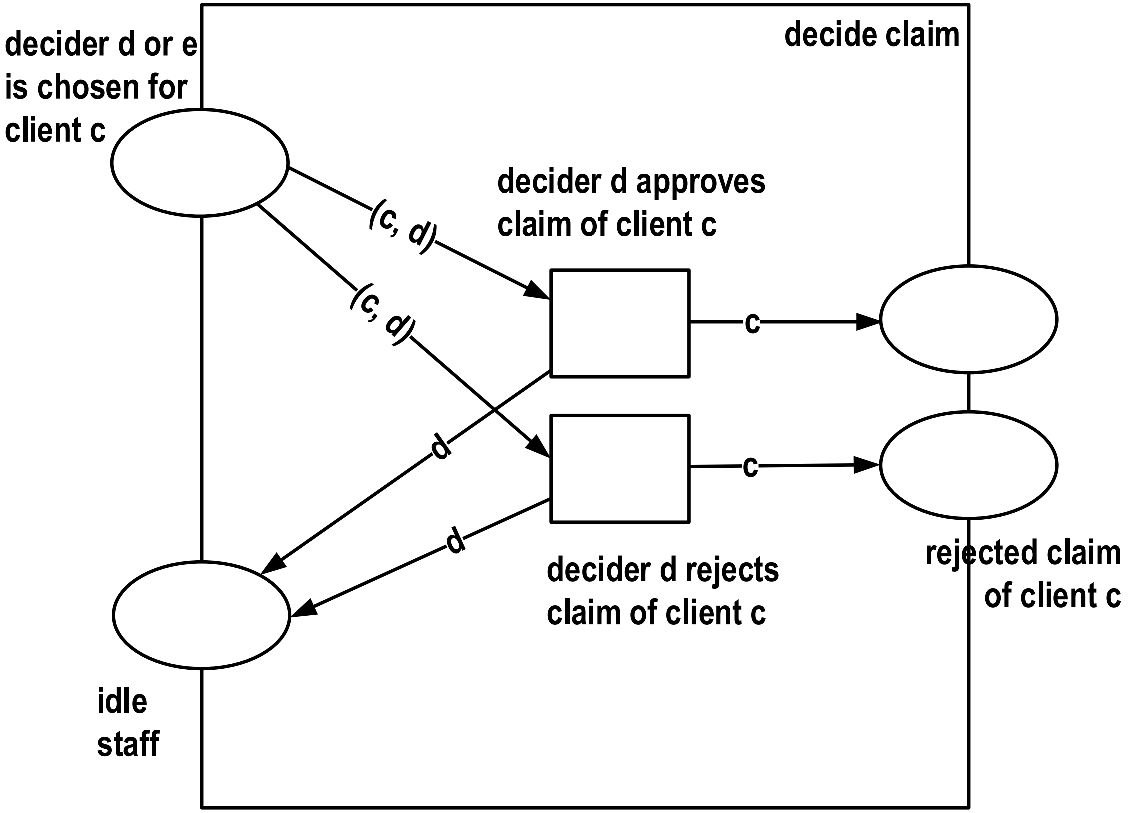




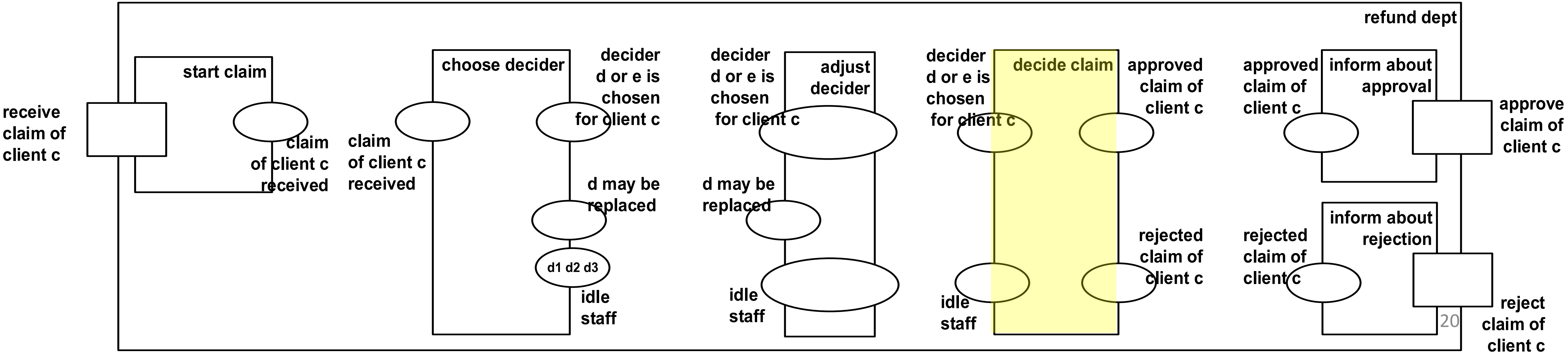
Part II system modules



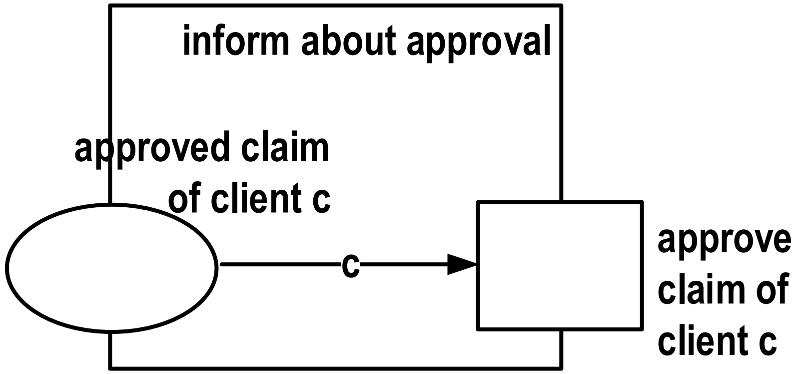


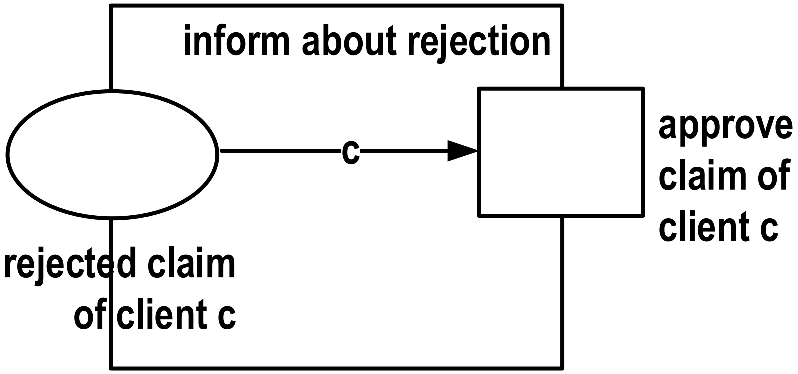


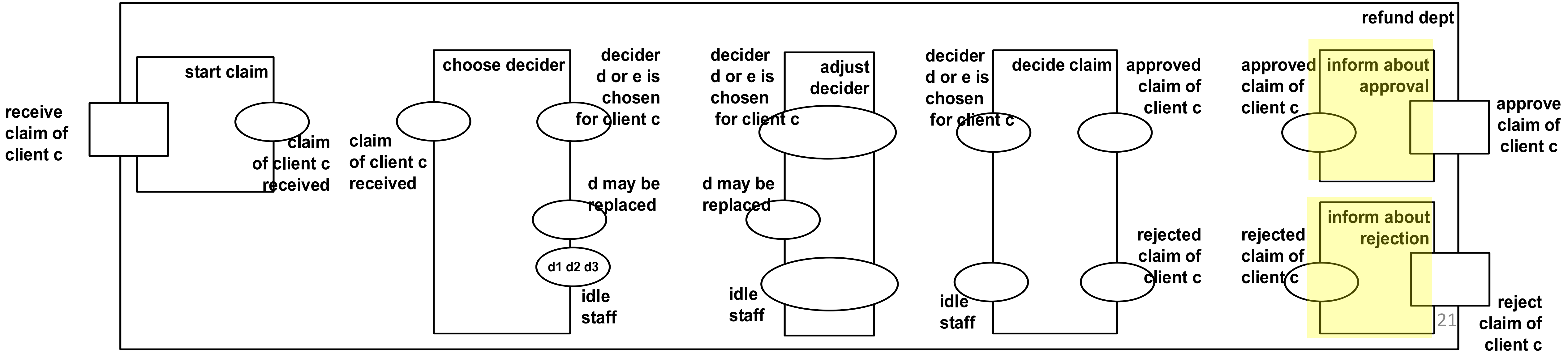
Part II system modules



Part II system modules

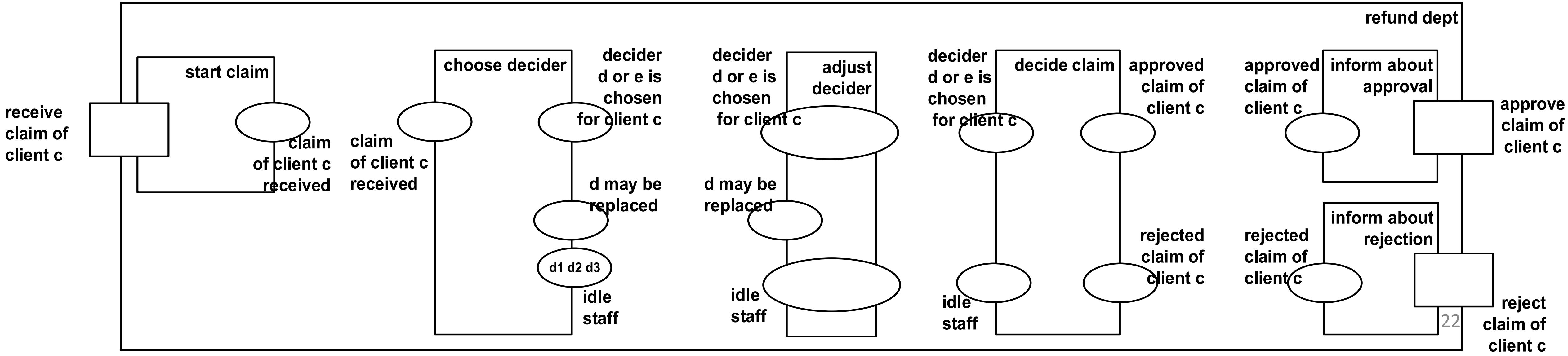






Part II system modules

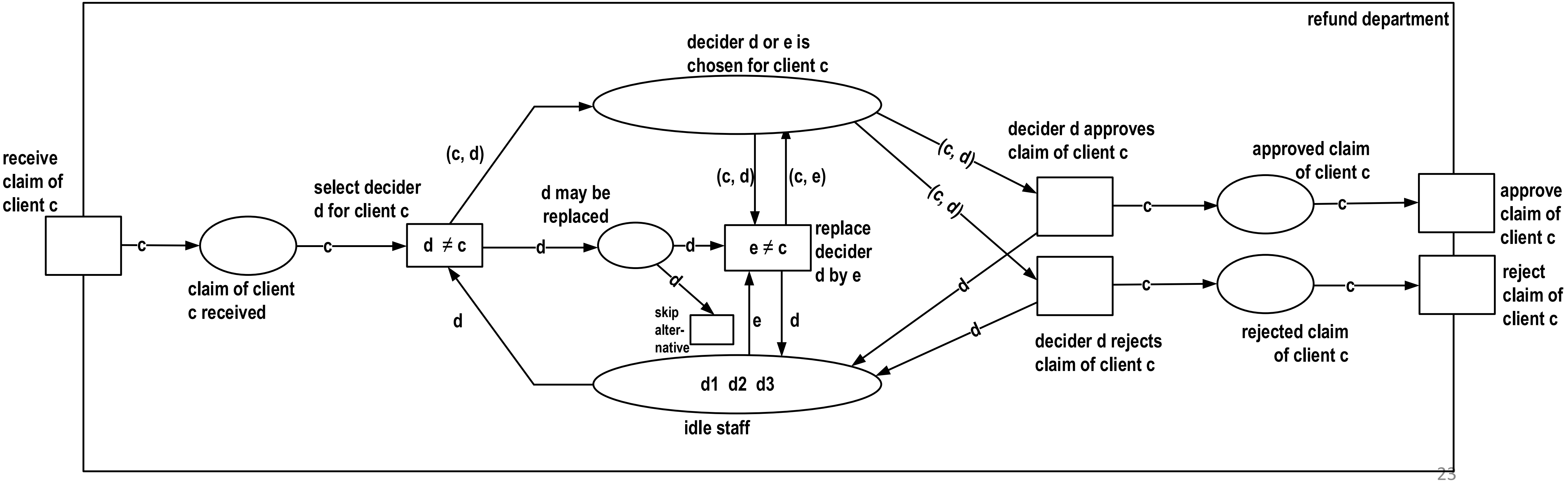
refund department =
start claim • choose decider • adjust decider • decide claim • inform about approval • inform about rejection



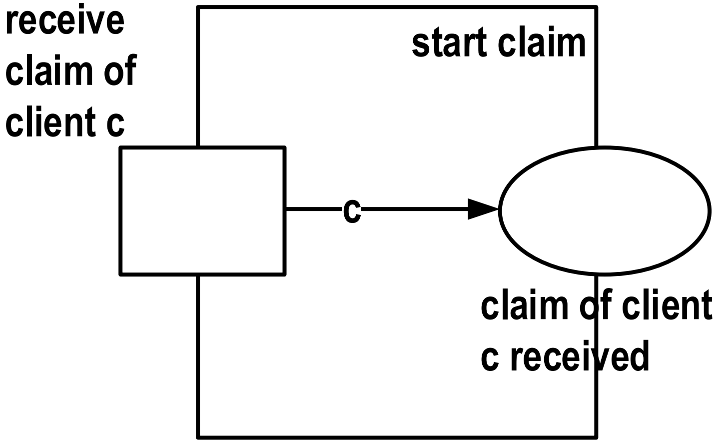
Part III liveness

refund department ⊨
occurs(receive claim of client c . c1) ↦
occurs (approve claim of client c. c1)
∨ occurs(reject claim of client c. c1)

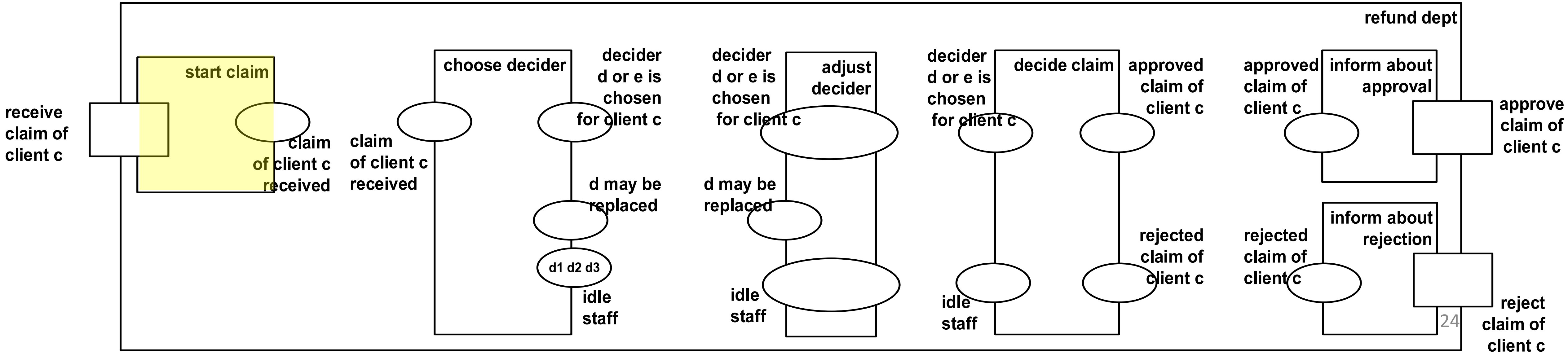
How to prove this property?
!Compositionally!



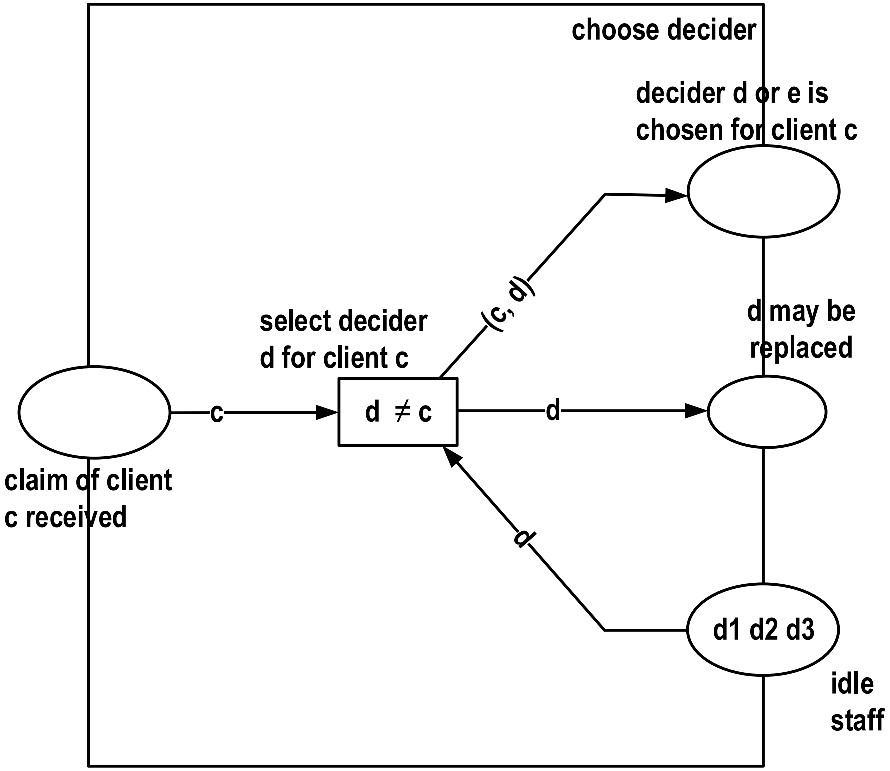
Part III liveness



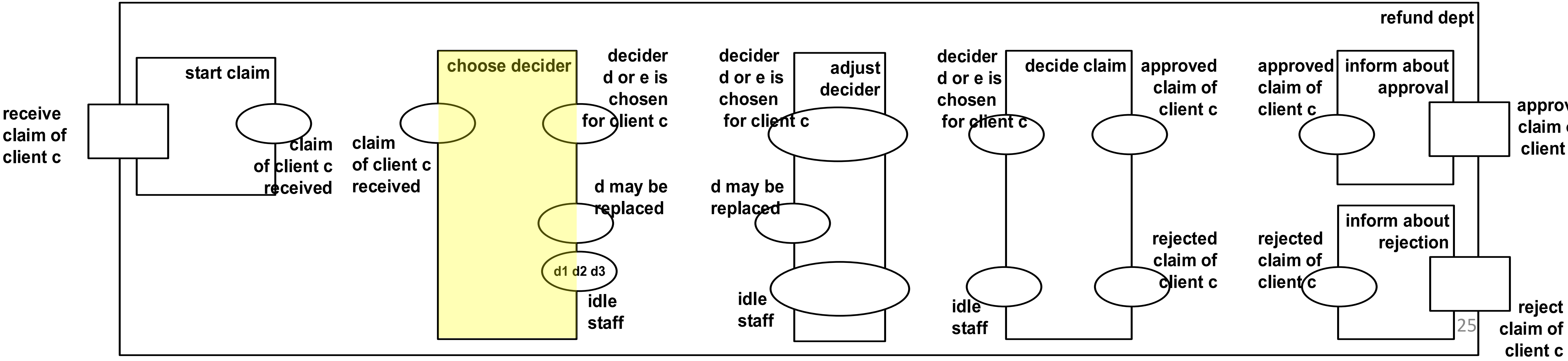
start claim ⊨
occurs(receive claim of client c . c1)
↦ claim of client c received.c1

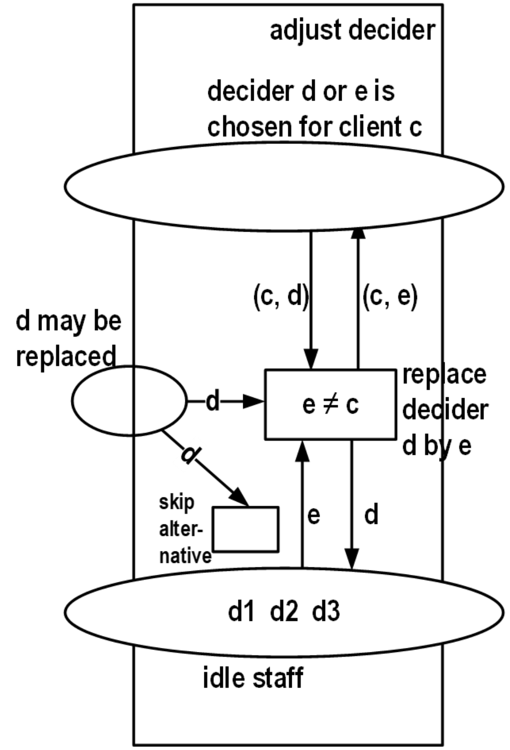


Part III liveness



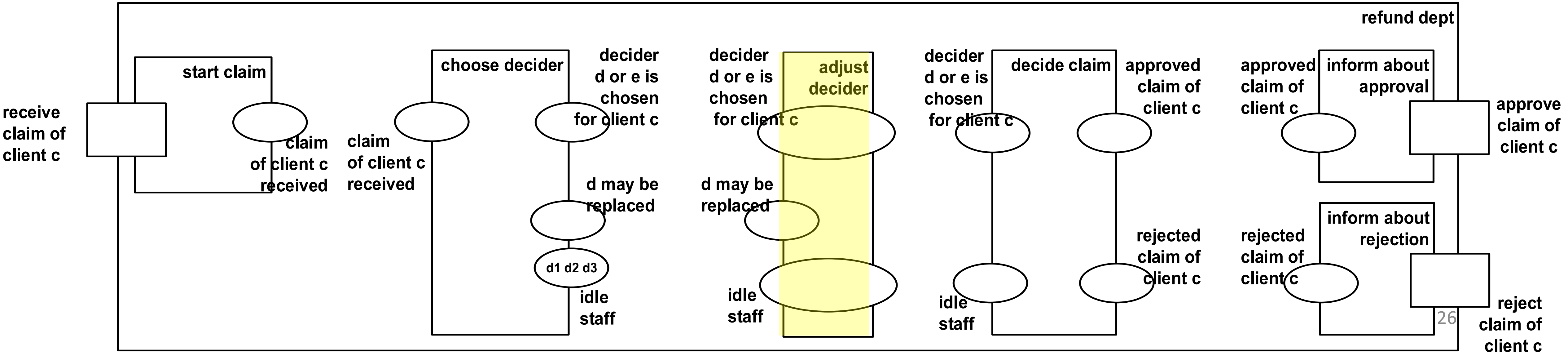
choose decider ⊨
idle staff.e ∧ claim of client c received.c1
↦
decider d or e is chosen for client c
∧ d may be replaced

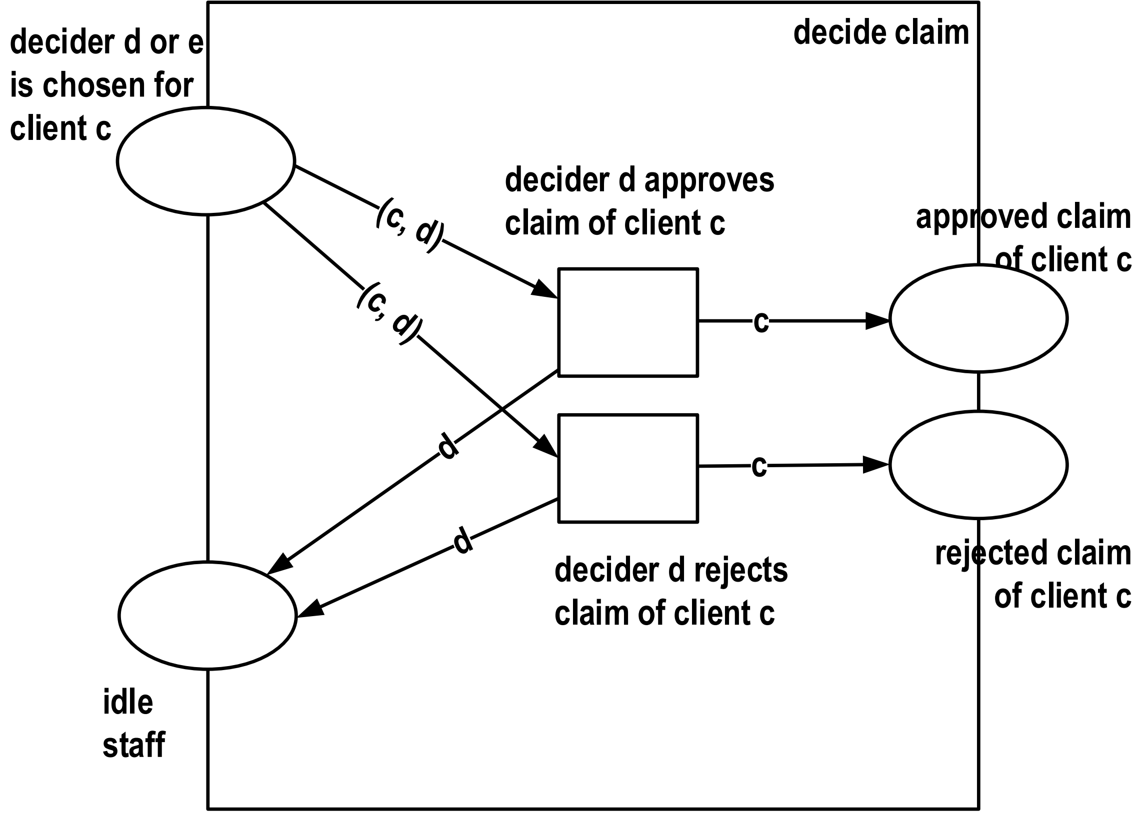




Part III Liveness

adjust decider ⊨
idle staff. d
∧ decider d or e is chosen for client c .d
∧ d may be replaced
↦
decider d or e is chosen for client c. d





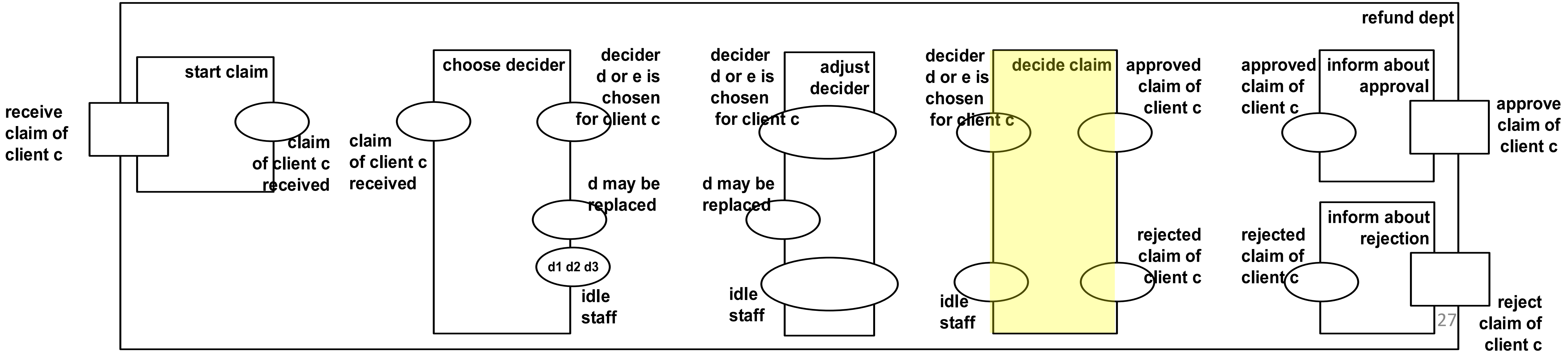
Part III liveness

decide claim ⊨

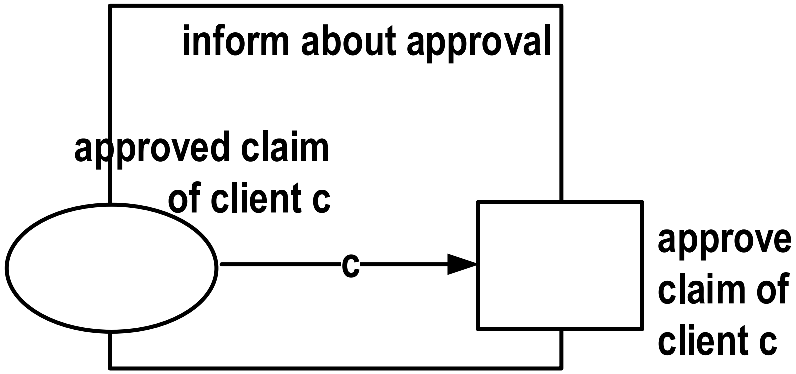
decider d or e is chosen for client c. d

↦ approved claim of client c.c1

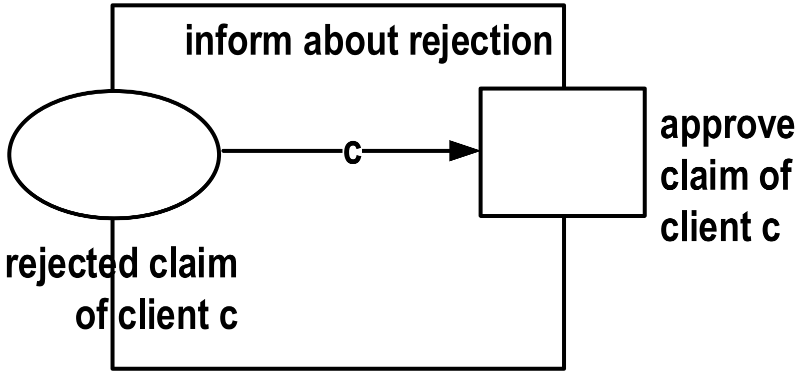
v rejected claim of client c.c1



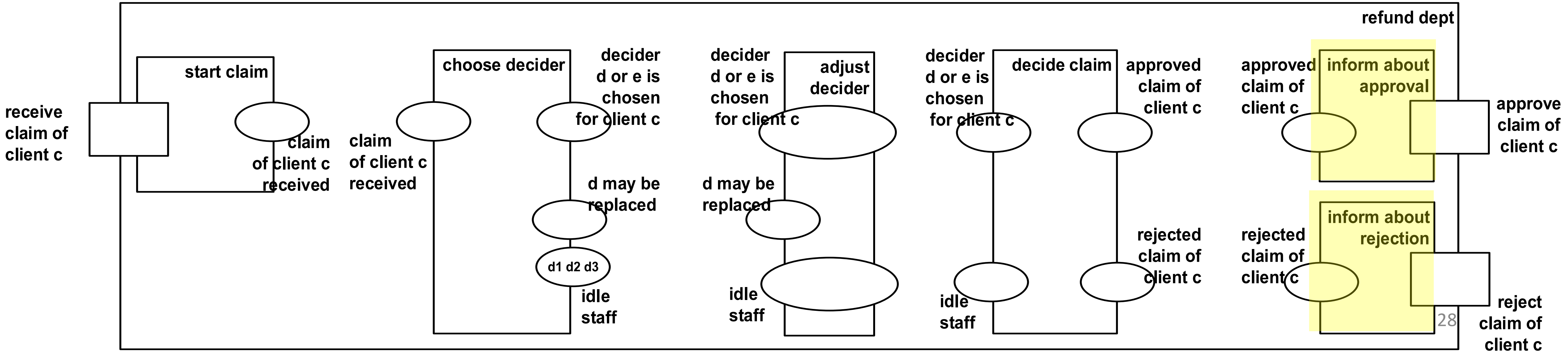
Part III liveness



inform about approval ⊨
approved claim of client c . c1
↦ occurs(approve claim of client c . c1)



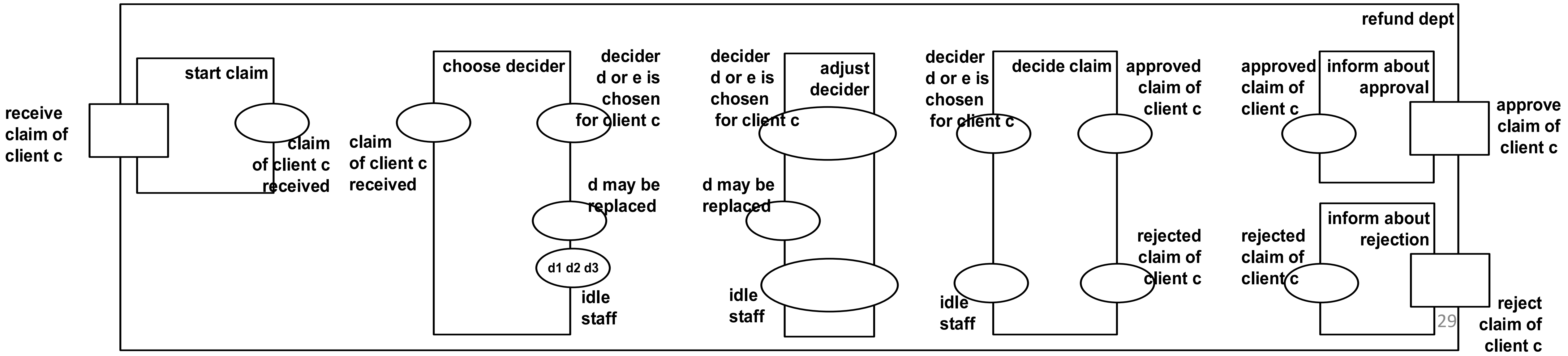
inform about rejection ⊨
rejected claim of client c . c1
↦ occurs(reject claim of client c . c1)



Part III liveness

Theorem Let M and N be two modules with M ⊨ Mp ↦ Mq and N ⊨ Np ↦ Nq.
Let Mq ↦ Np.
Then M • N ⊨ Mp ↦ Nq.
... cum grano salis

refund department ⊨
occurs(receive claim of client c . c1) ↦
occurs (approve claim of client c. c1)
∨ occurs(reject claim of client c. c1)



What did we gain?

- Modular Composition of modules

analogy:

words u, v, w over an alphabet Σ:

w can be composed at two ends: $u \bullet w$ and $w \bullet v$

Furthermore, $(u \bullet w) \bullet v = u \bullet (w \bullet v)$,

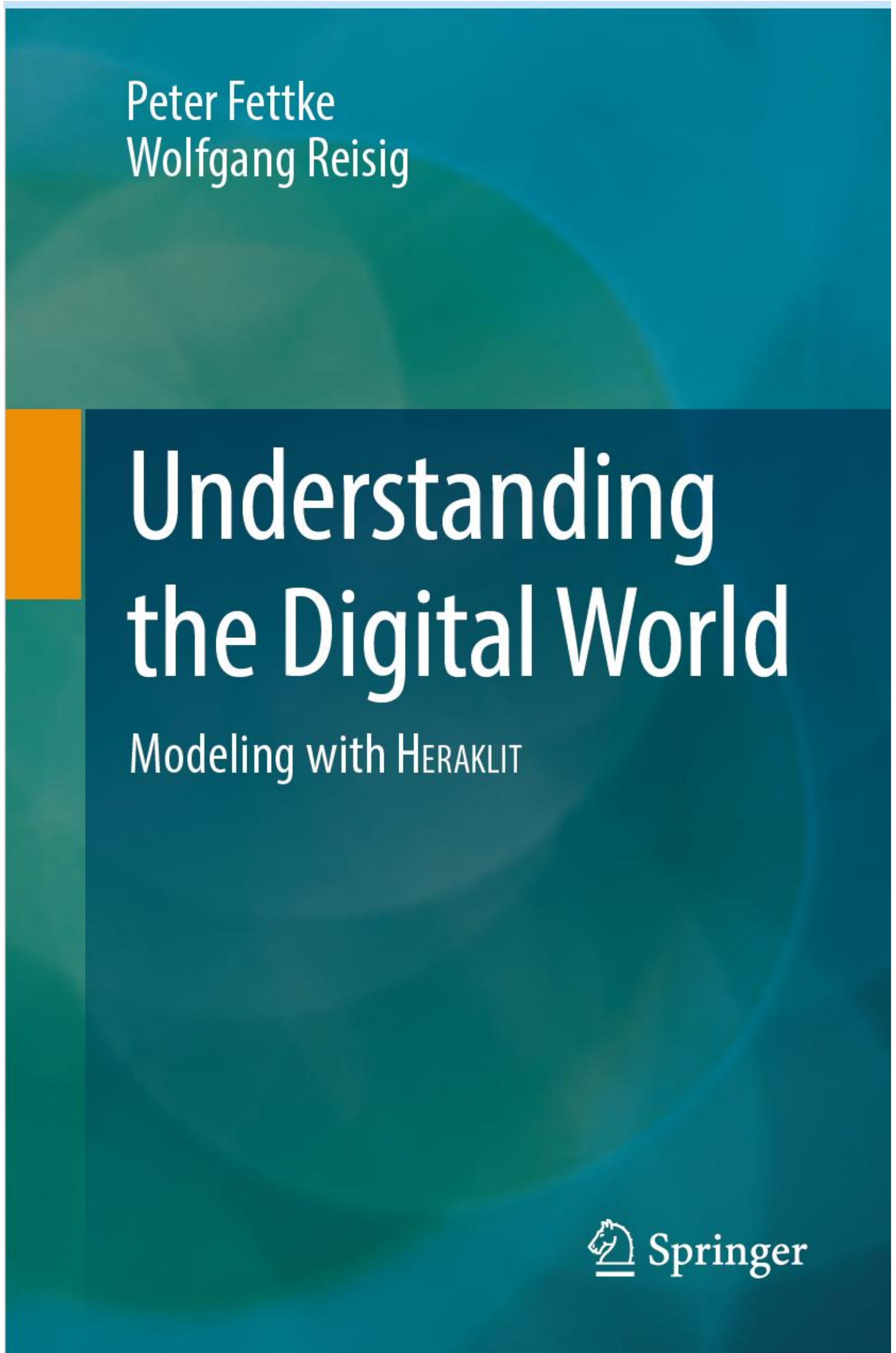
written $u \bullet w \bullet v$ or just uwv

Replace words with modules!

What did we gain?

- Modularization, Refinement, abstraction
- universal, expressive composition
  total, deterministic, associative
- an algebraic representation of modules
- modular analysis techniques

for details, see



SummerSoc 2026 Χερσονισος, Ελλαδα

Trust by design
– in praise of modularization: a case study

the end

Wolfgang Reisig

Humboldt-Universität zu Berlin Germany

Peter Fettke

German Research Center for Artificial Intelligence (DFKI) and Saarland University, Saarbrücken, Germany

Institut für Informatik
Humboldt-Universität zu Berlin

Deutsches Forschungszentrum für Künstliche Intelligenz
German Research Center for Artificial Intelligence

UNIVERSITÄT DES SAARLANDES