MULAN4SONAR – A Tutorial

Michael Köhler-Bußmeier

Department for Informatics
University of Hamburg

June 13, 2012

Agent vs. Organisation; Service vs. System

Agents

- local autonomy
- inteligence:
  goal-orientation, learning
- social awareness:
  cooperation, competition

Problems

- bounded rationality:
  sub-optimal solutions
- scaling problem for large agent populations

Organisations

- global coherence
- organisational learning,
  data mining
- teams

Solutions

- social structures:
  roles, protocols, norms
- reduction of complexity:
  interaction networks,
  e.g. hierarchies

Agent vs. Organisation; Service vs. System

Agents

- local autonomy
- inteligence:
  goal-orientation, learning
- social awareness:
  cooperation, competition

Problems

- bounded rationality:
  sub-optimal solutions
- scaling problem for large agent populations

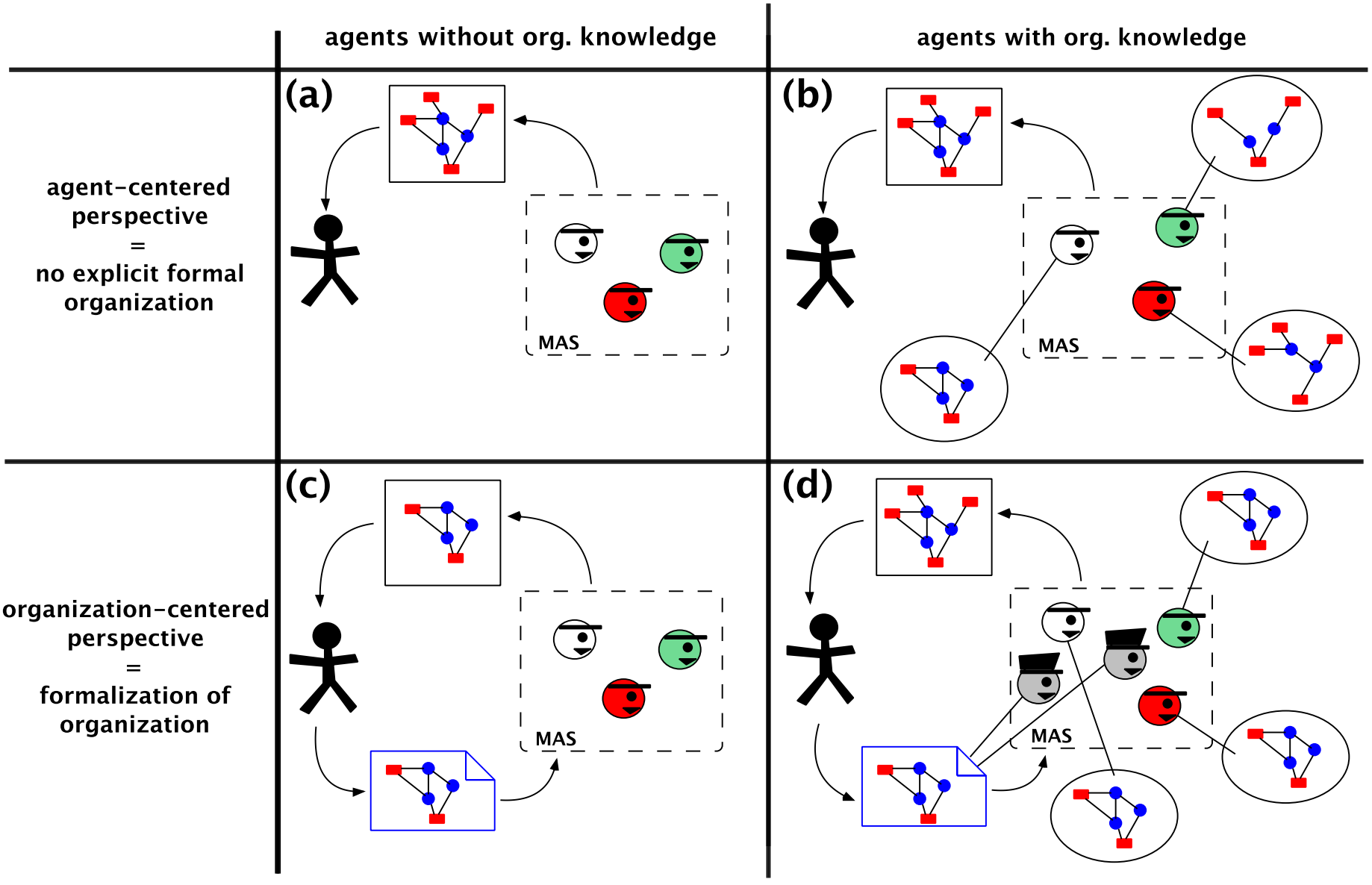
Organisations

- global coherence
- organisational learning,
  data mining
- teams

Solutions

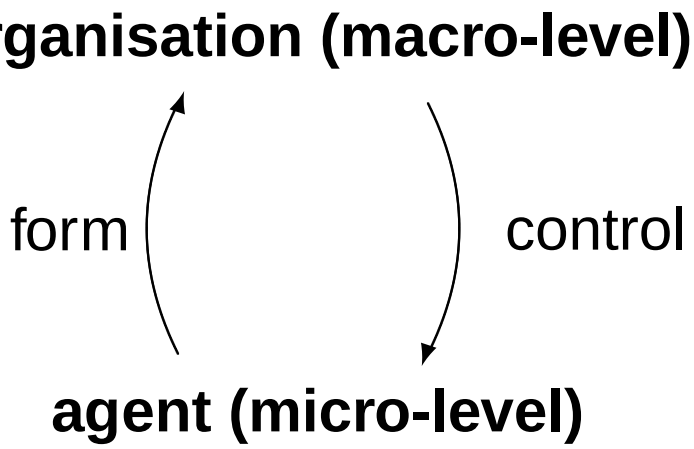
- social structures:
  roles, protocols, norms
- reduction of complexity:
  interaction networks,
  e.g. hierarchies

Matrix: Organisation Awareness



Duality of the Concept ’Organisation’

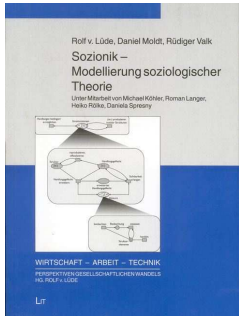
Cyclic dependencies of micro (agents) and macro (organisation):



Sociology: *middle-up-down* instead of *top-down*/*bottom-up*

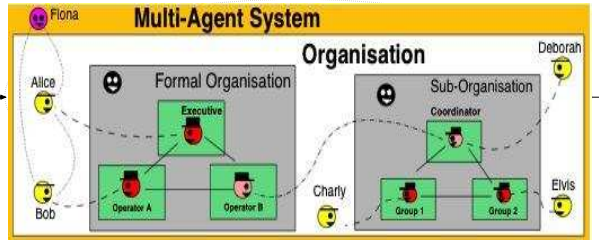
⇒ Organisations as modelling components and sw-artefacts

DFG-SPP Socionics:
Modells of Social Theories



Overview of SONAR Concepts

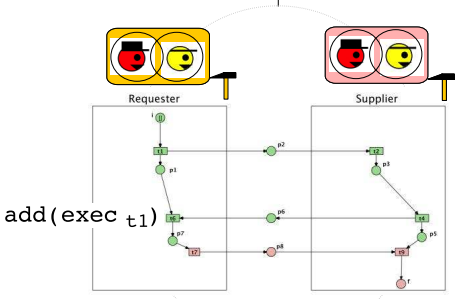
reorganise



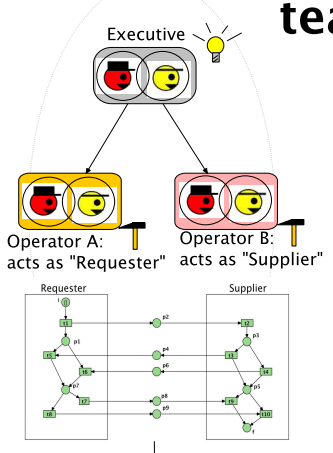
form team

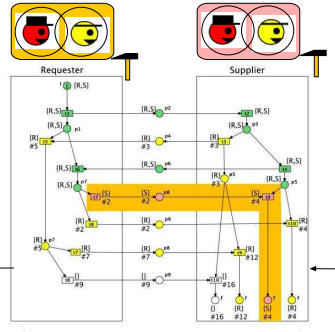
Organsiation-MAS

transformation



team





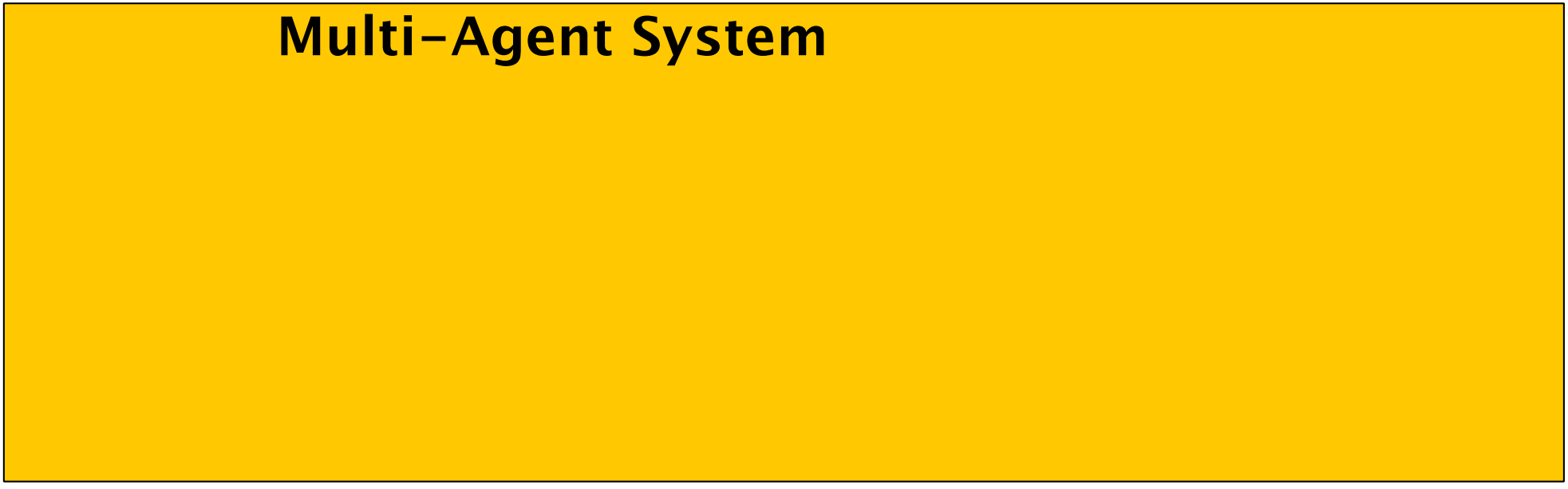
team plan

execute

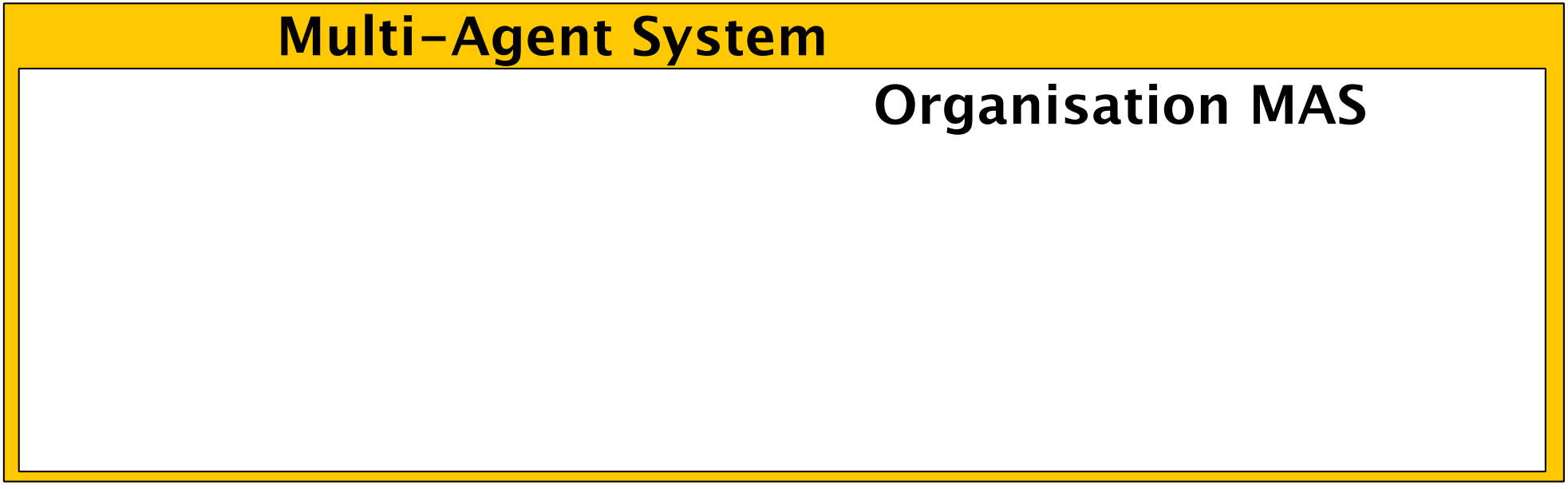
negotiate

Outline

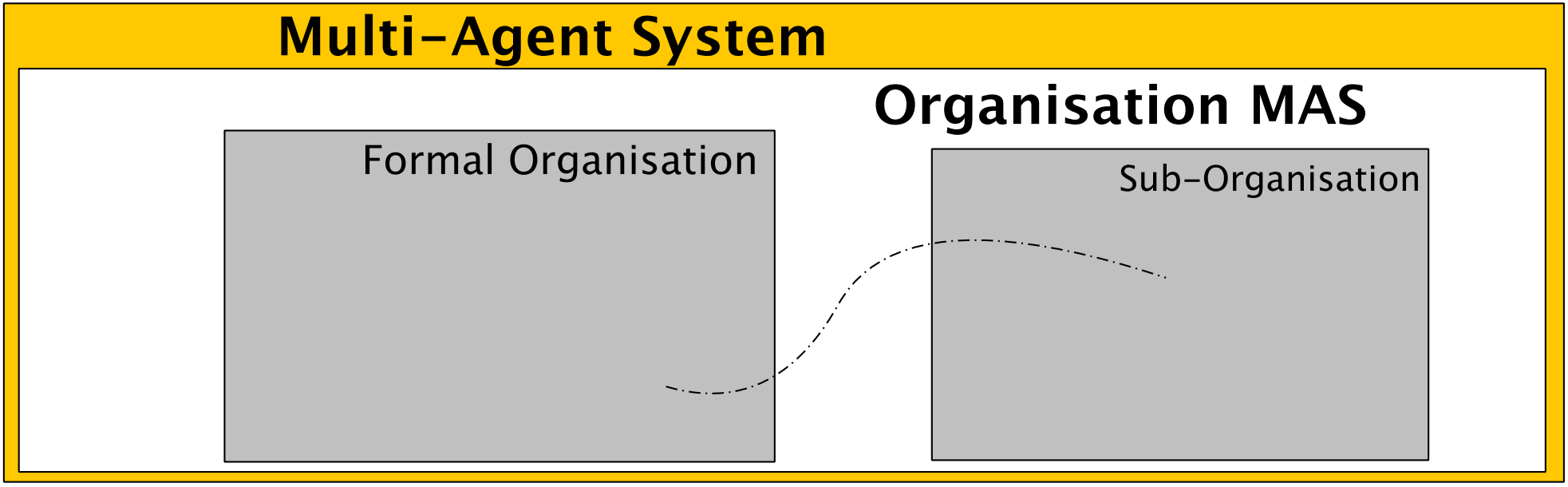
The MULAN4SONAR Organisation MAS



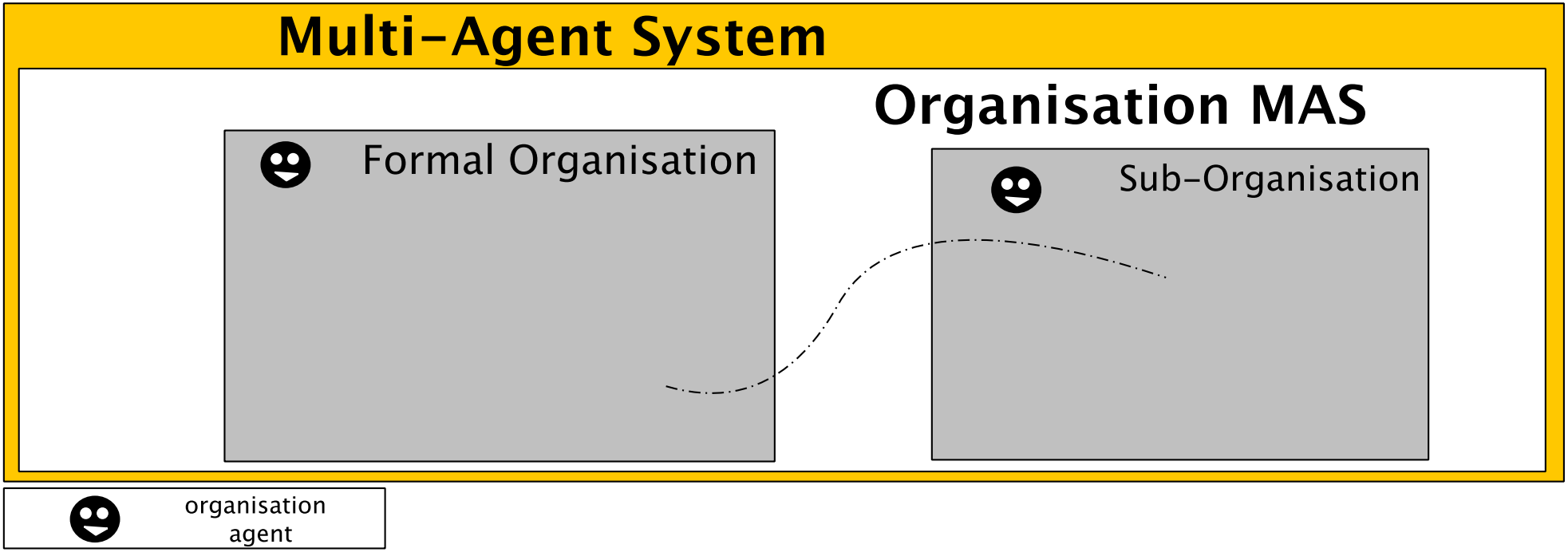
The MULAN4SONAR Organisation MAS



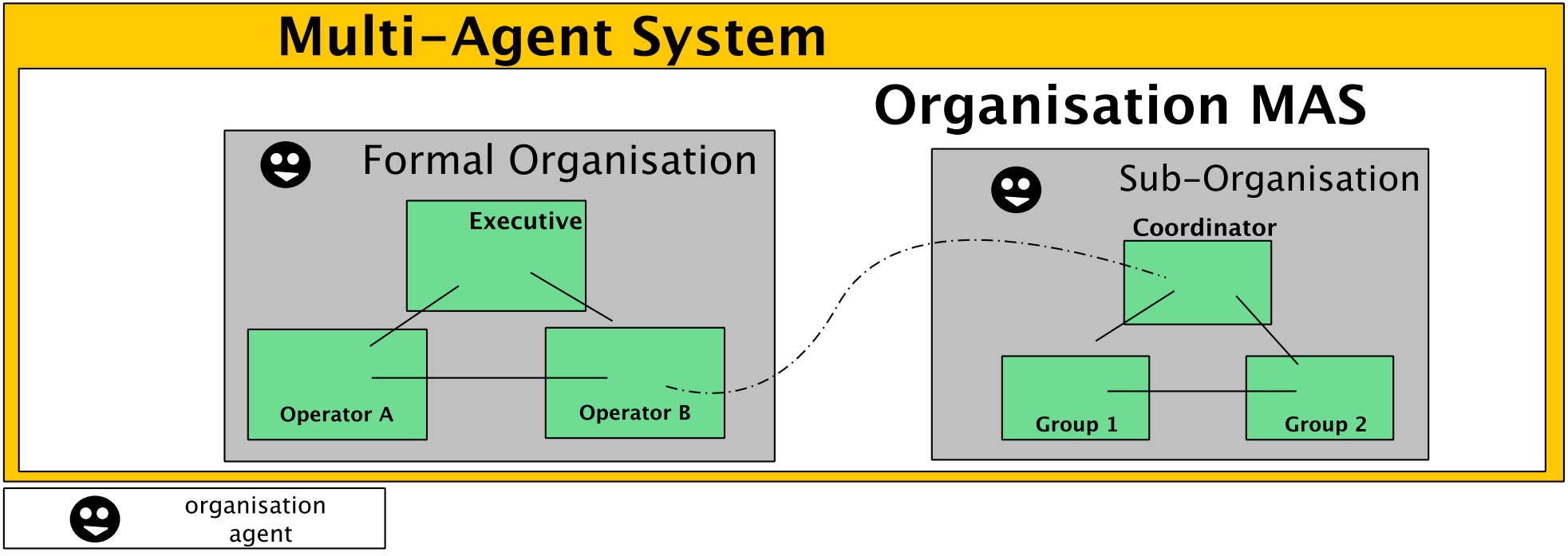
The MULAN4SONAR Organisation MAS



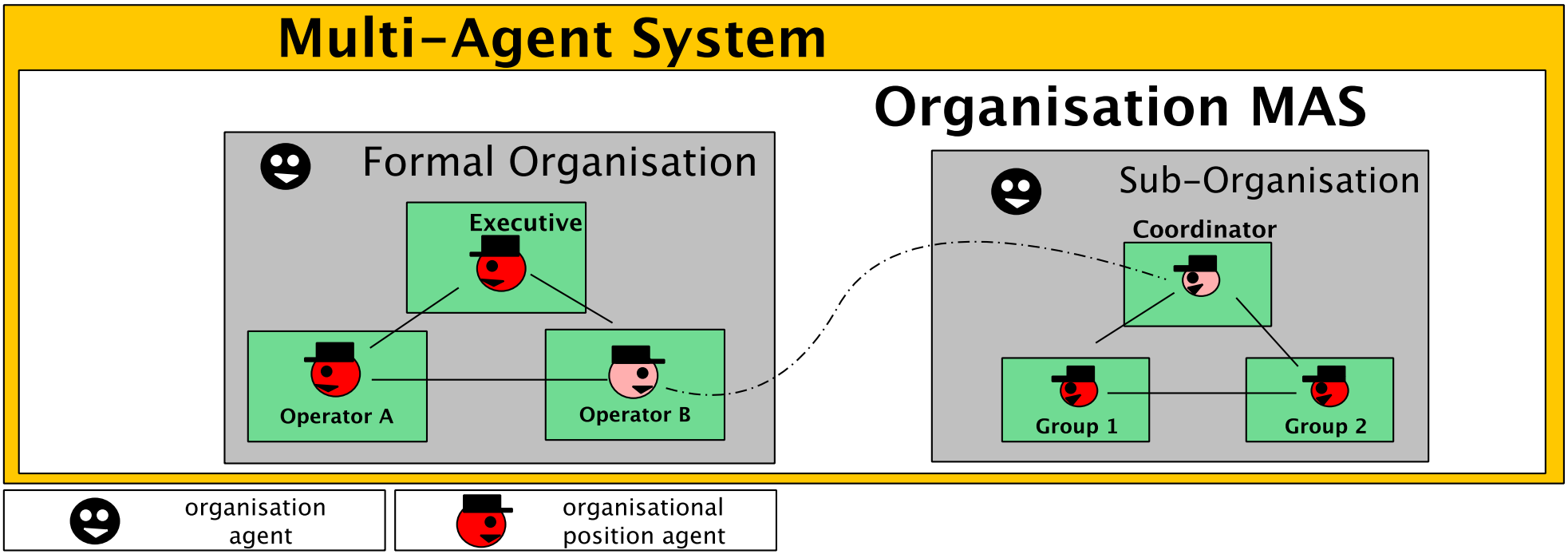
The MULAN4SONAR Organisation MAS



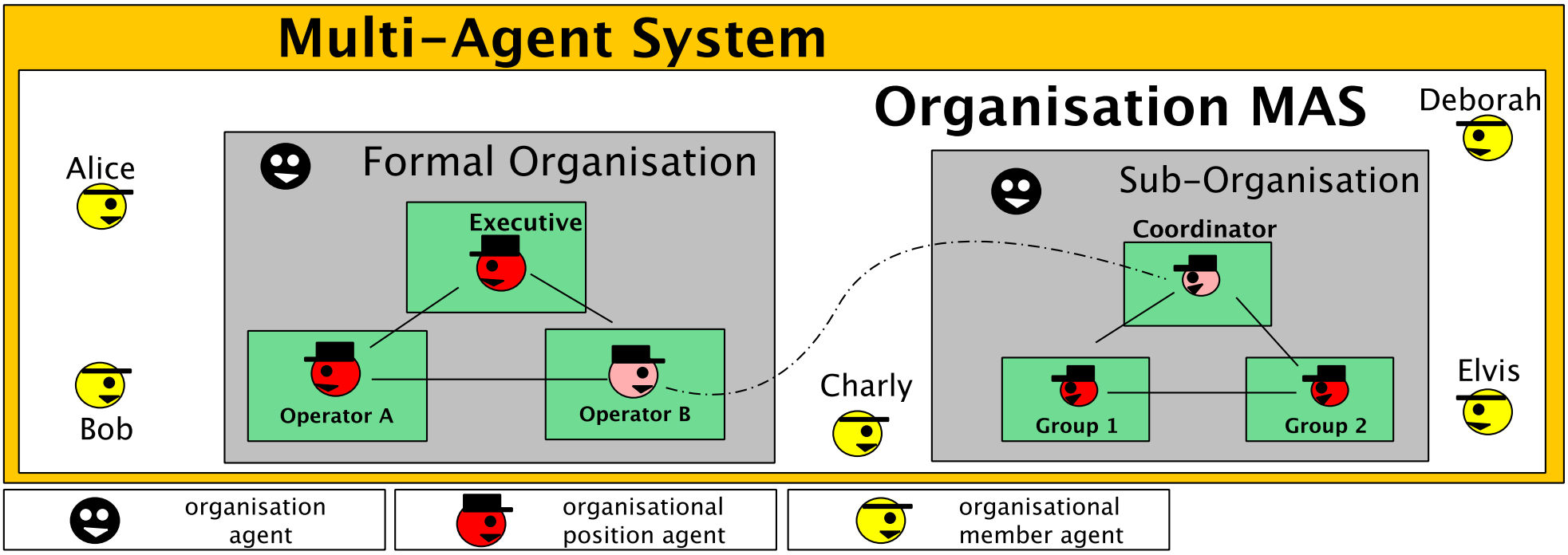
The MULAN4SONAR Organisation MAS



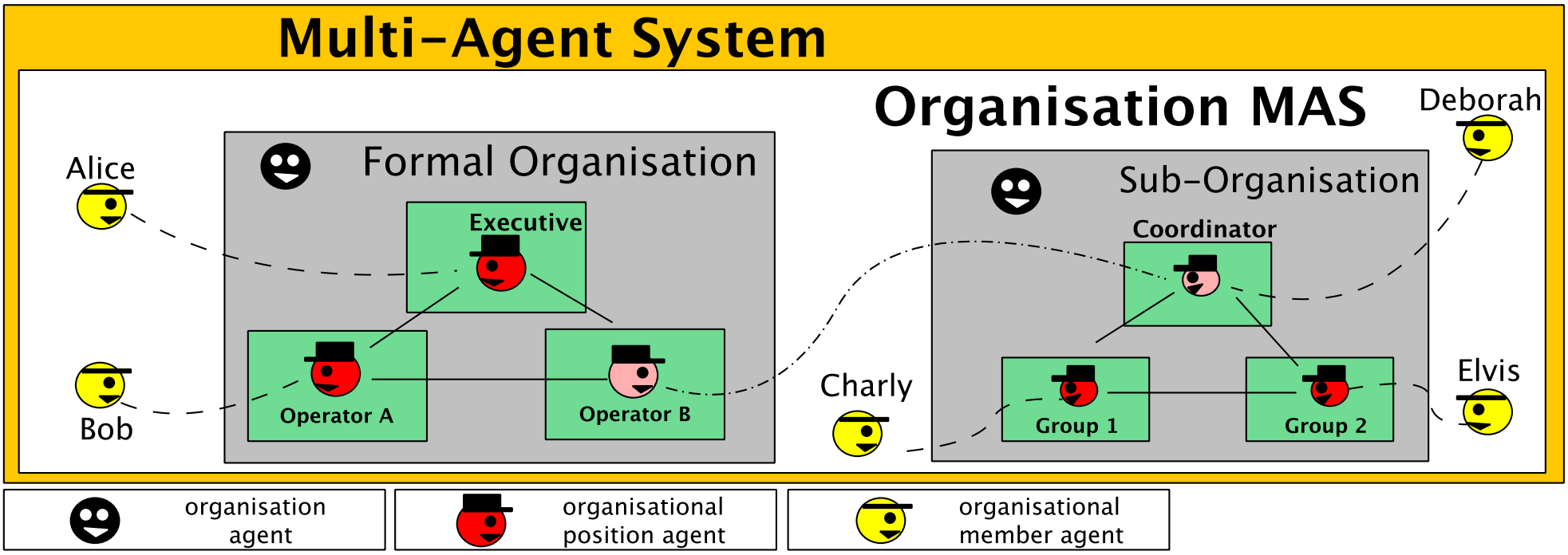
The MULAN4SONAR Organisation MAS



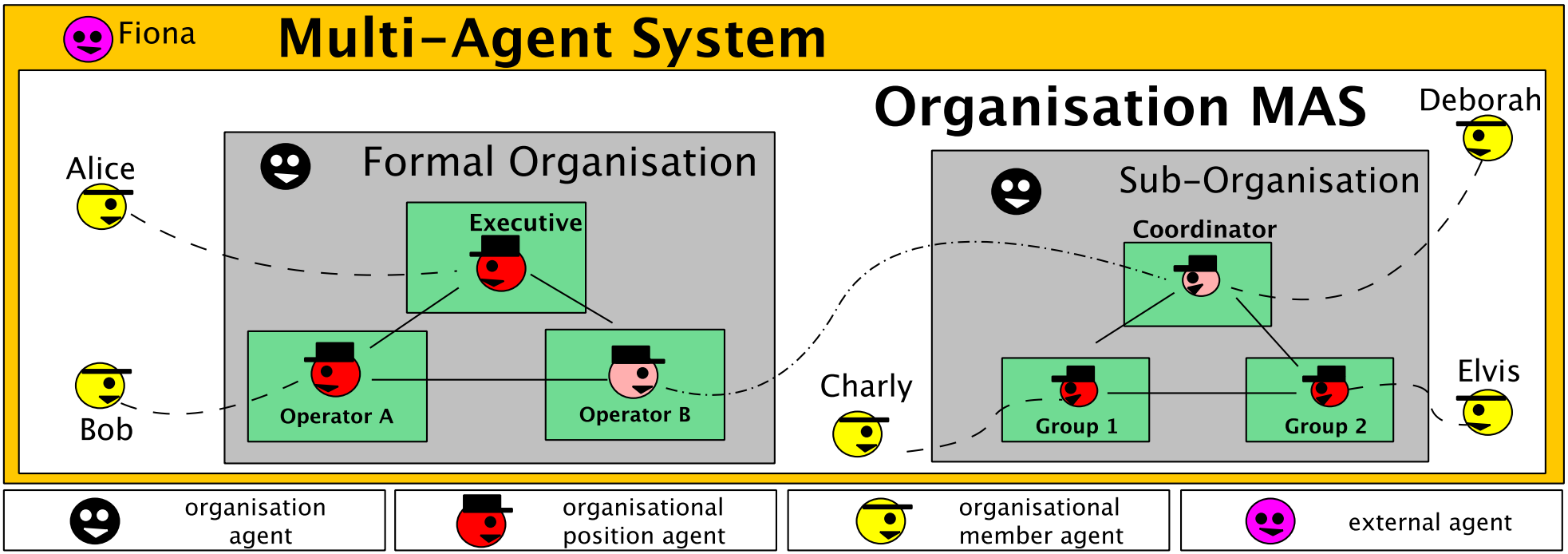
The MULAN4SONAR Organisation MAS



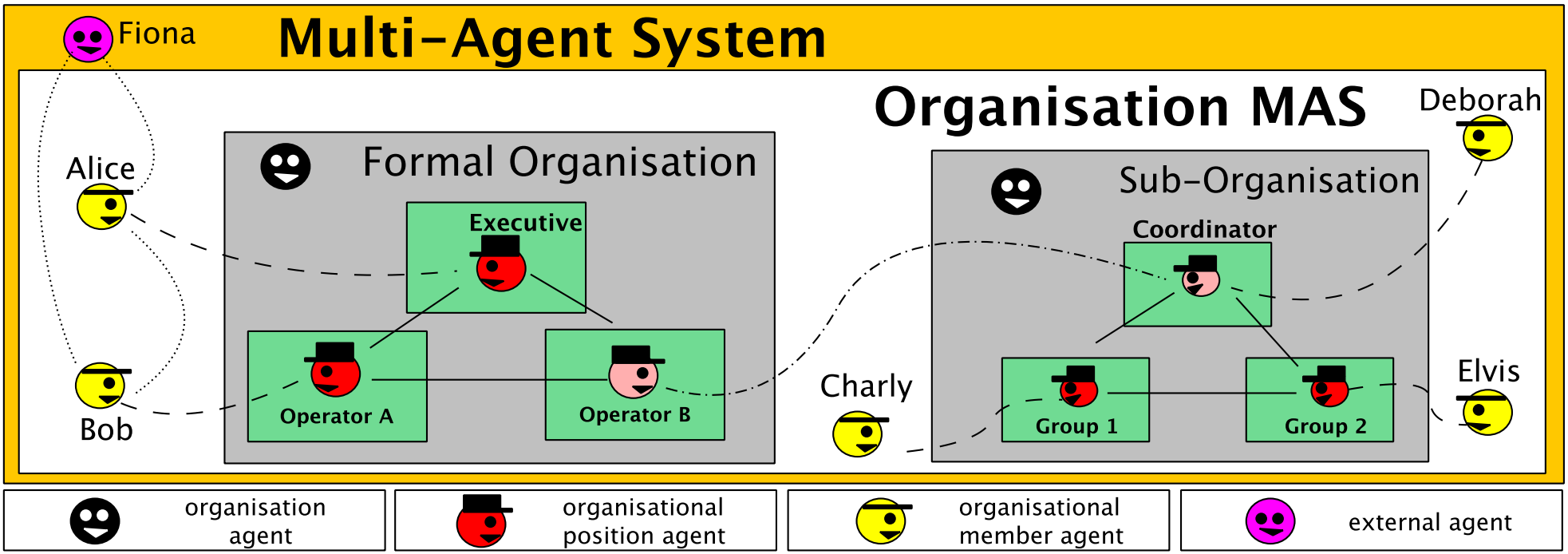
The MULAN4SONAR Organisation MAS



The MULAN4SONAR Organisation MAS

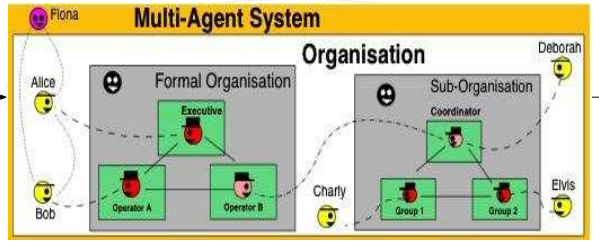


The MULAN4SONAR Organisation MAS



Overview of SONAR Concepts

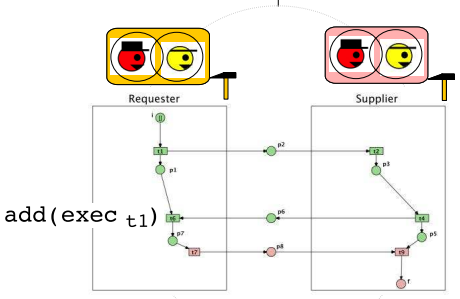
reorganise



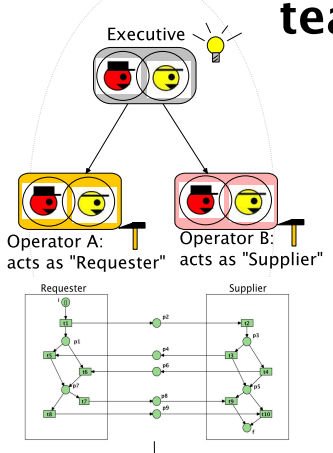
form team

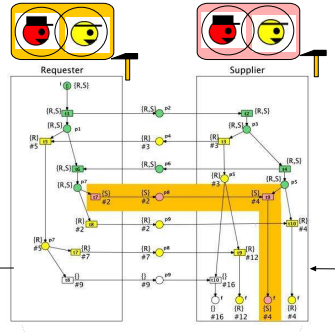
Organsiation-MAS

transformation



team





team plan

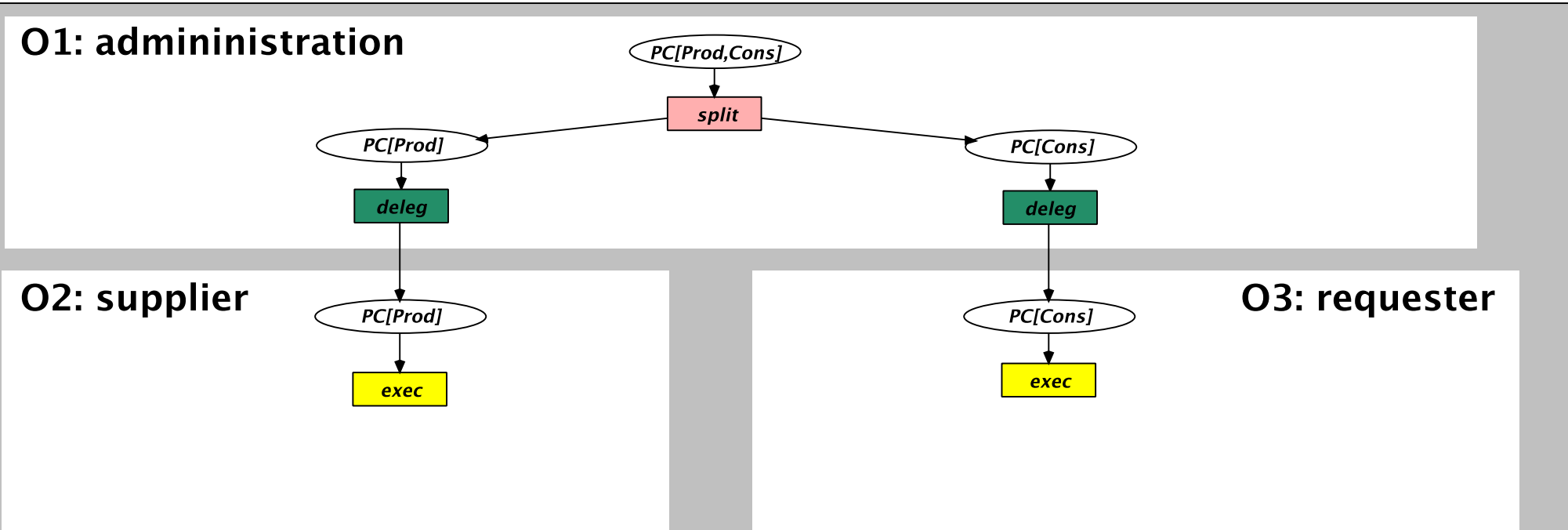
execute

negotiate

Outline

SONAR Organisation Models

- An **organisation** is a Petri net $N = (P, T, F)$.
- Places $\texttt{task}^{a}_{D[R]}$: Agent a has to implement the role R in the interaction D.
- Transitions: Implementation of tasks via ...
  1. delegation
  2. split
  3. refinement or
  4. execution



The SONAR Organisation Model

Let $\mathcal{T}$ be the set of all task implementations (i.e. delegations, split, refinement, and execution).

Definition

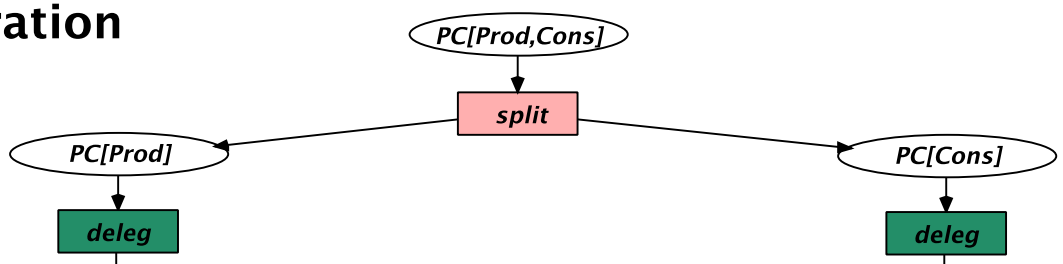
An **organisation** is a Petri net $N = (P, T, F)$ with

- $T \subseteq \mathcal{T}$,
- $P = {}^\bullet T \cup T^\bullet$, and
- $F = \mathcal{F} \cap (P \cup T)^2$.

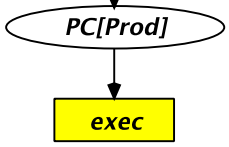
The places in $P^0 := \{p \in P \mid {}^\bullet p = \emptyset\}$ model those tasks that are generated by the environment.

Transformations on SONAR-Models

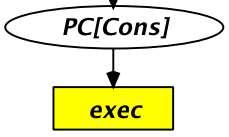
O1: admininistration



O2: supplier



O3: requester



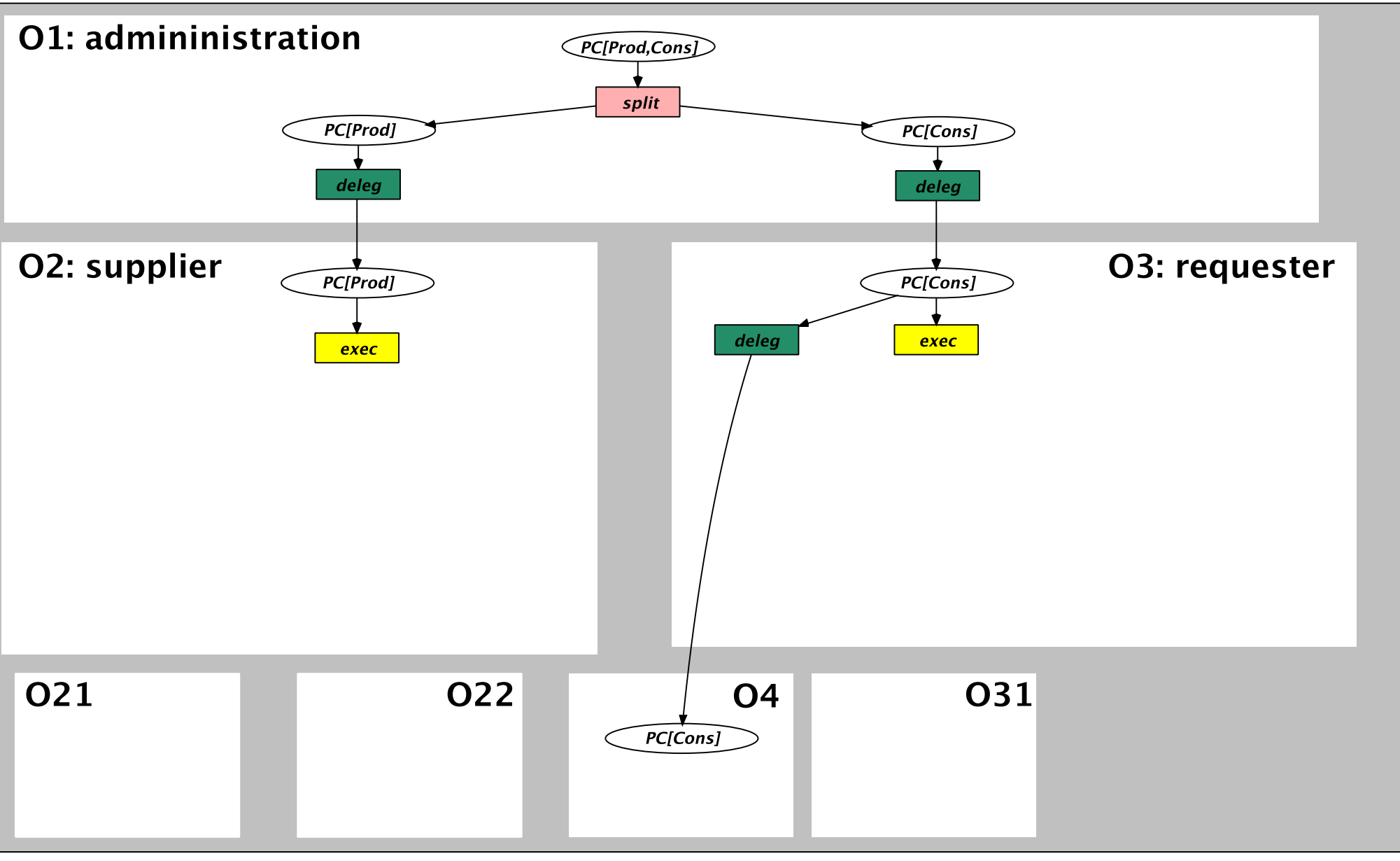
O21

O22

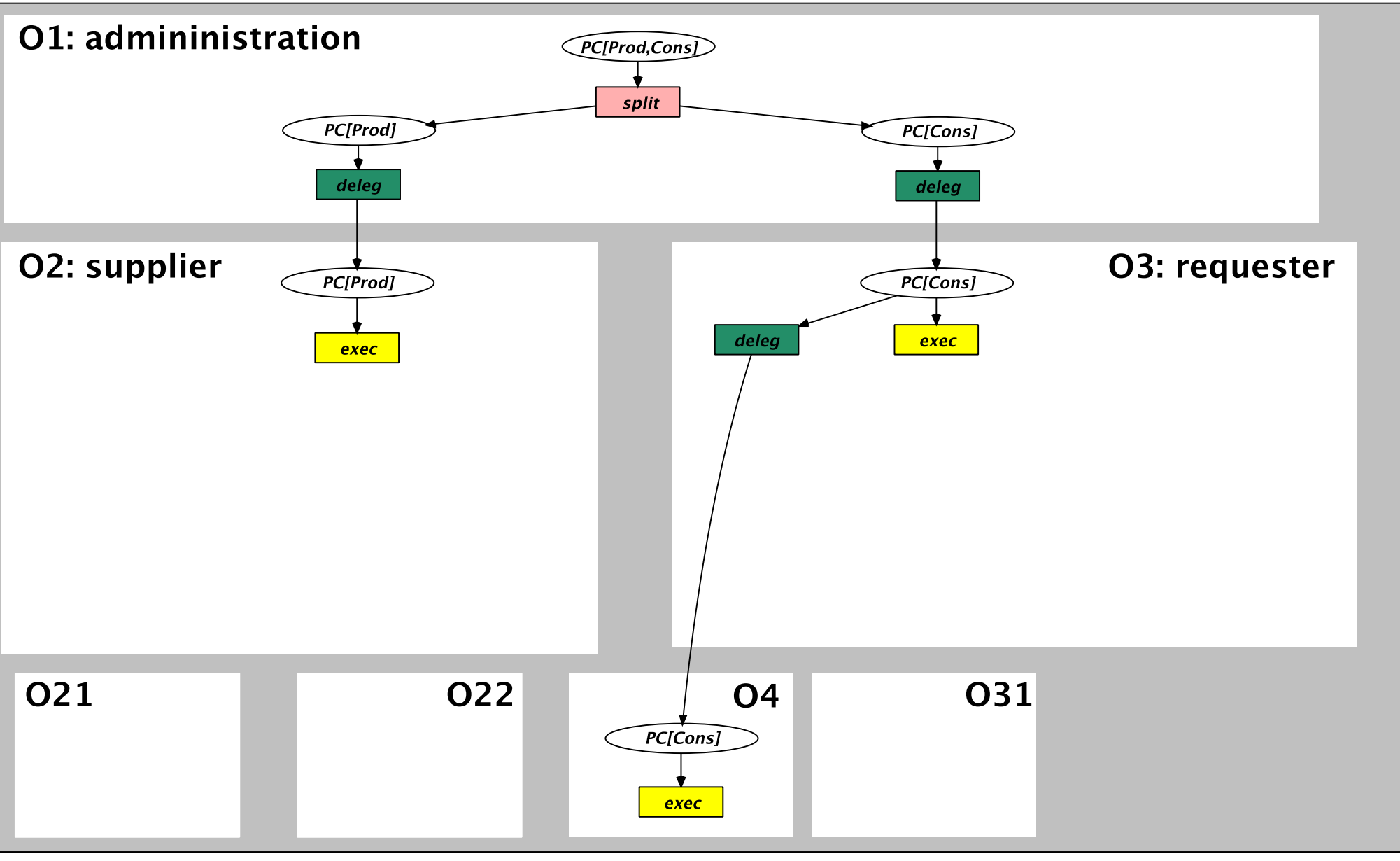
O4

O31

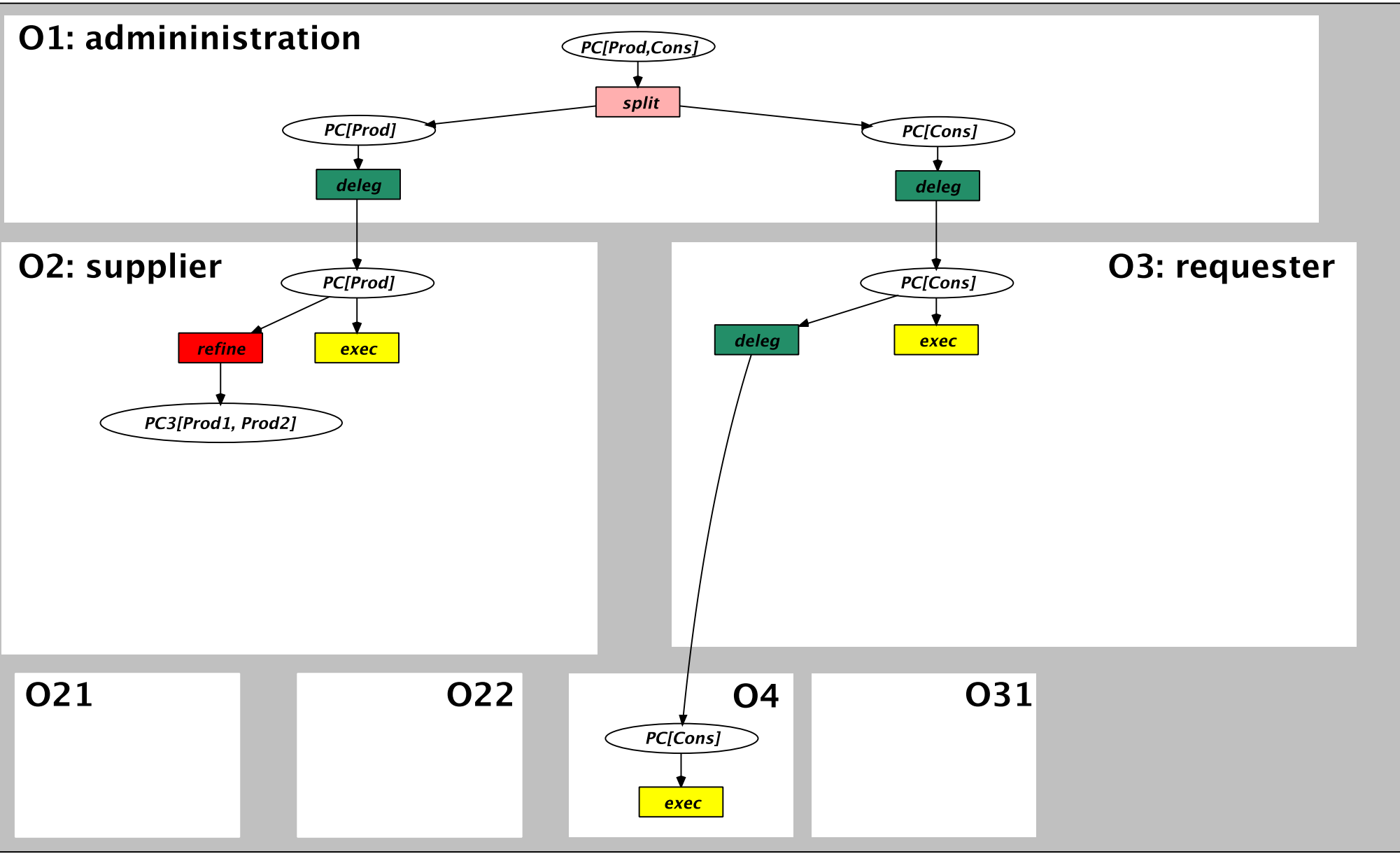
Transformations on SONAR-Models



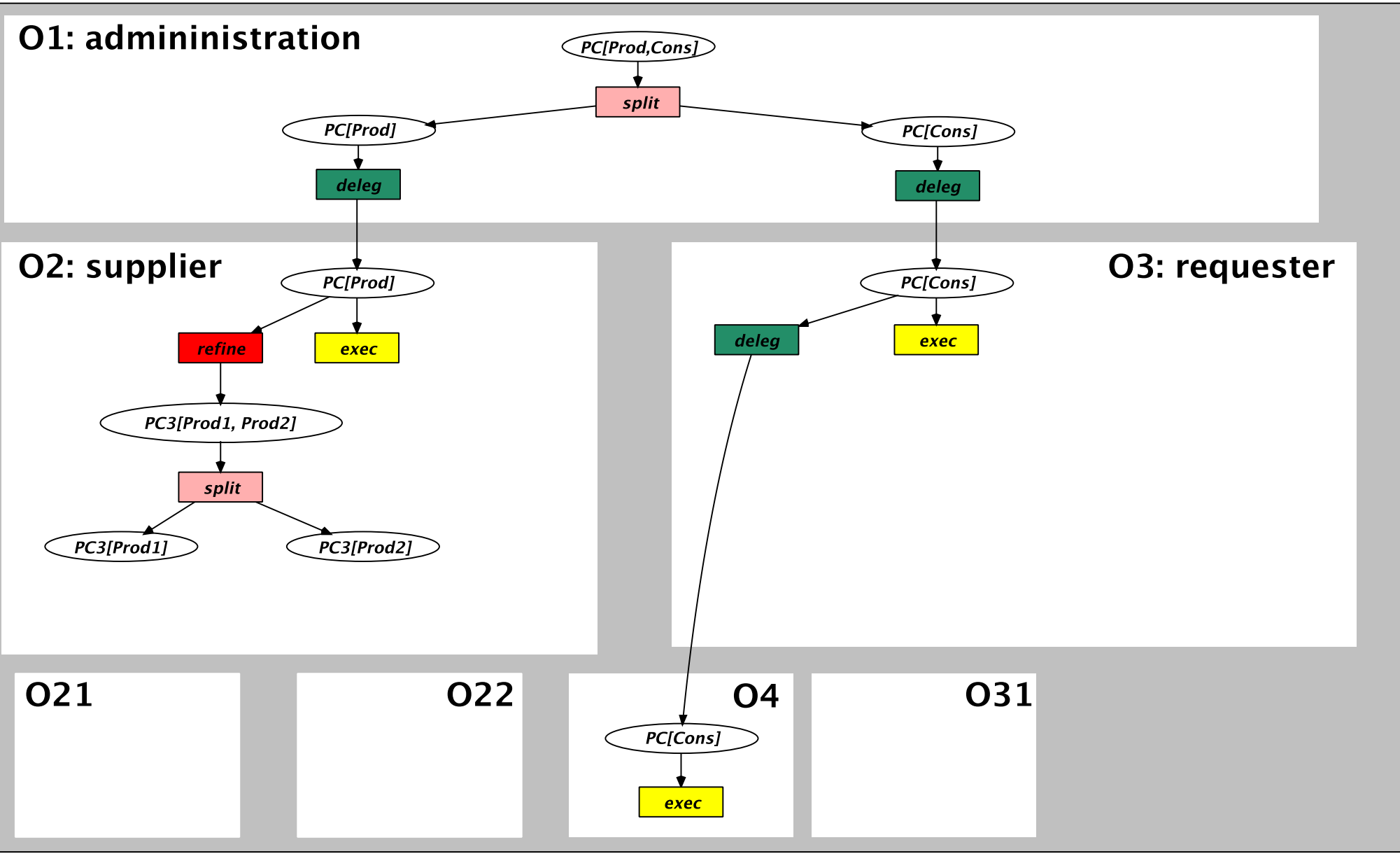
Transformations on SONAR-Models



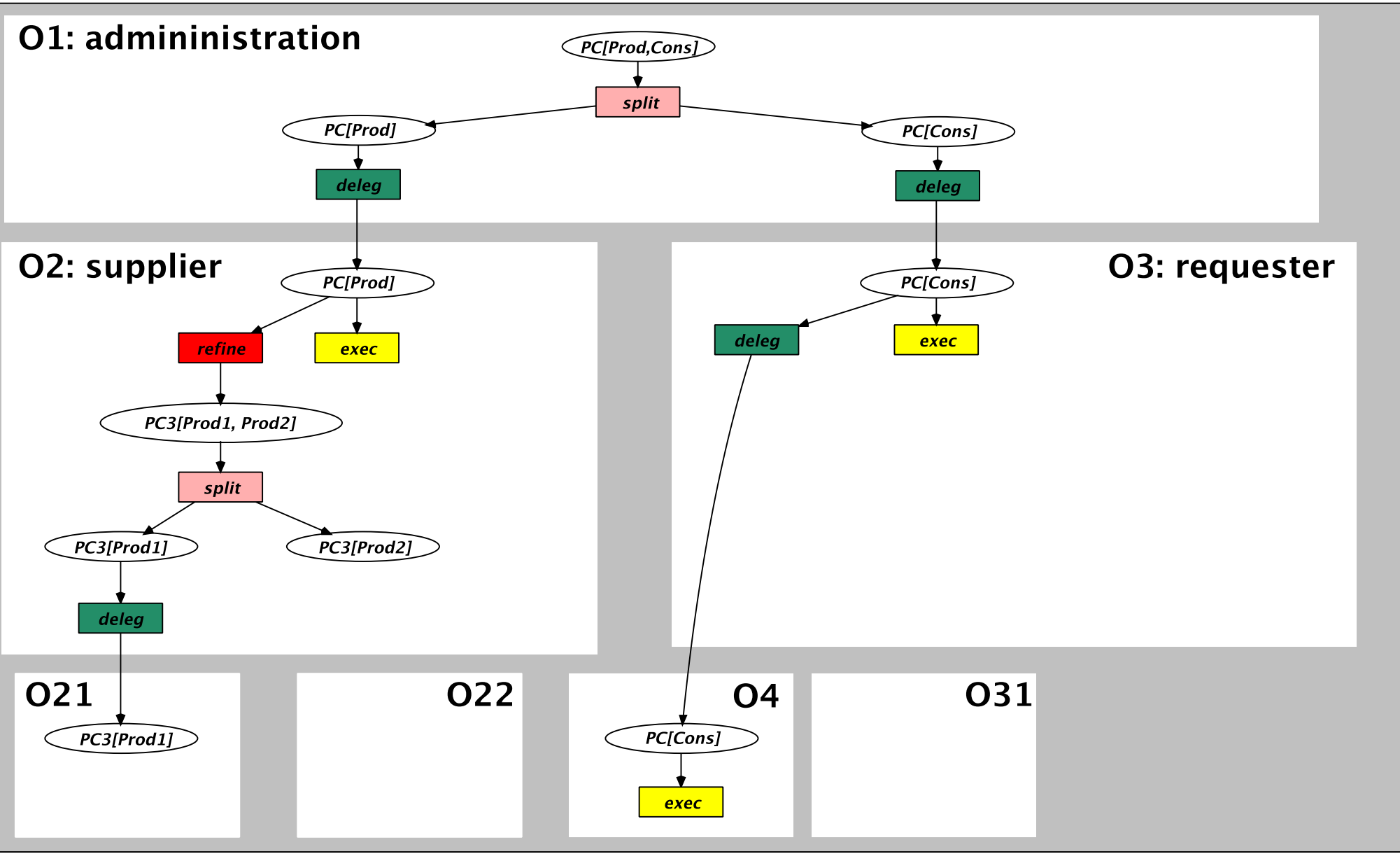
Transformations on SONAR-Models



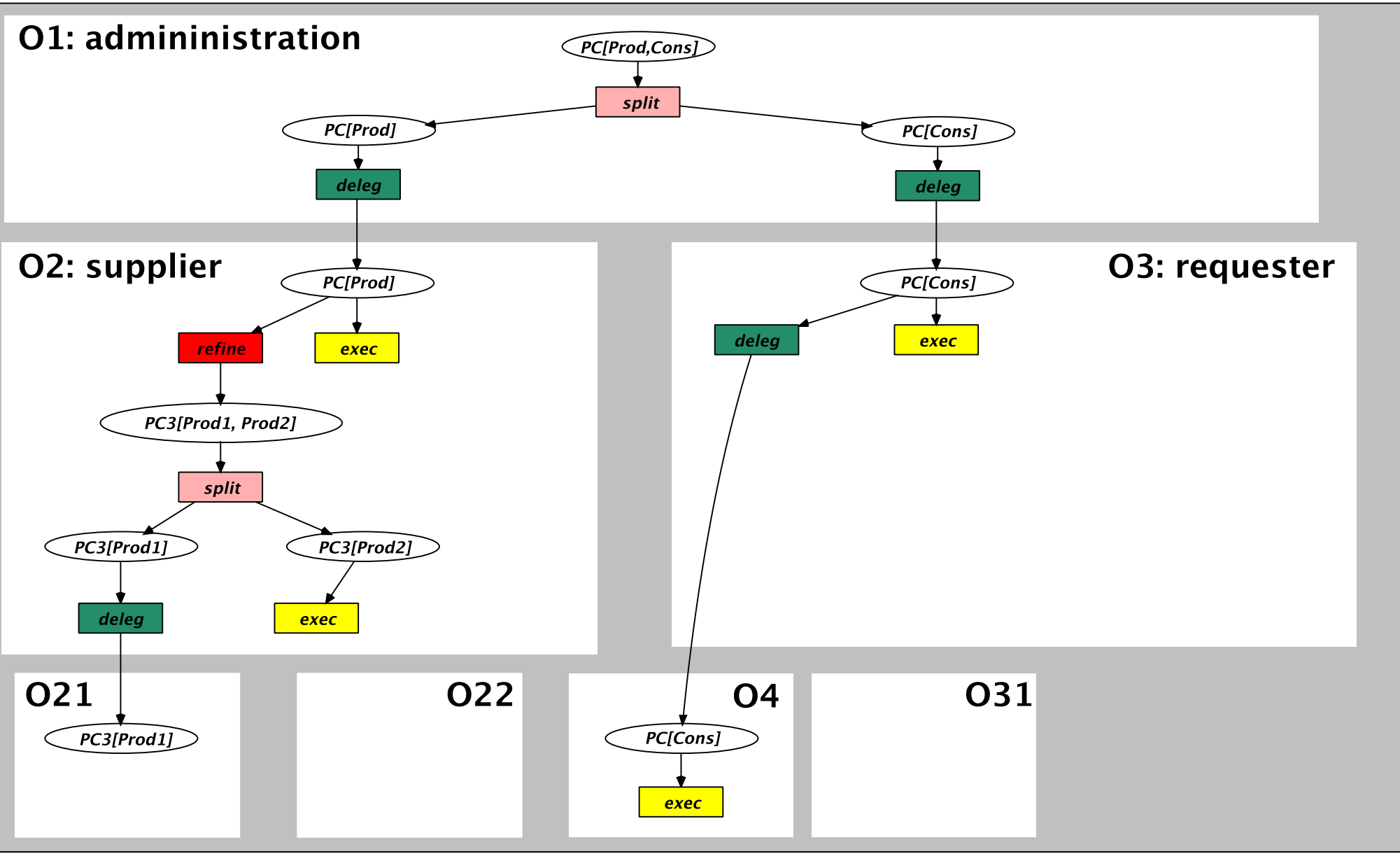
Transformations on SONAR-Models



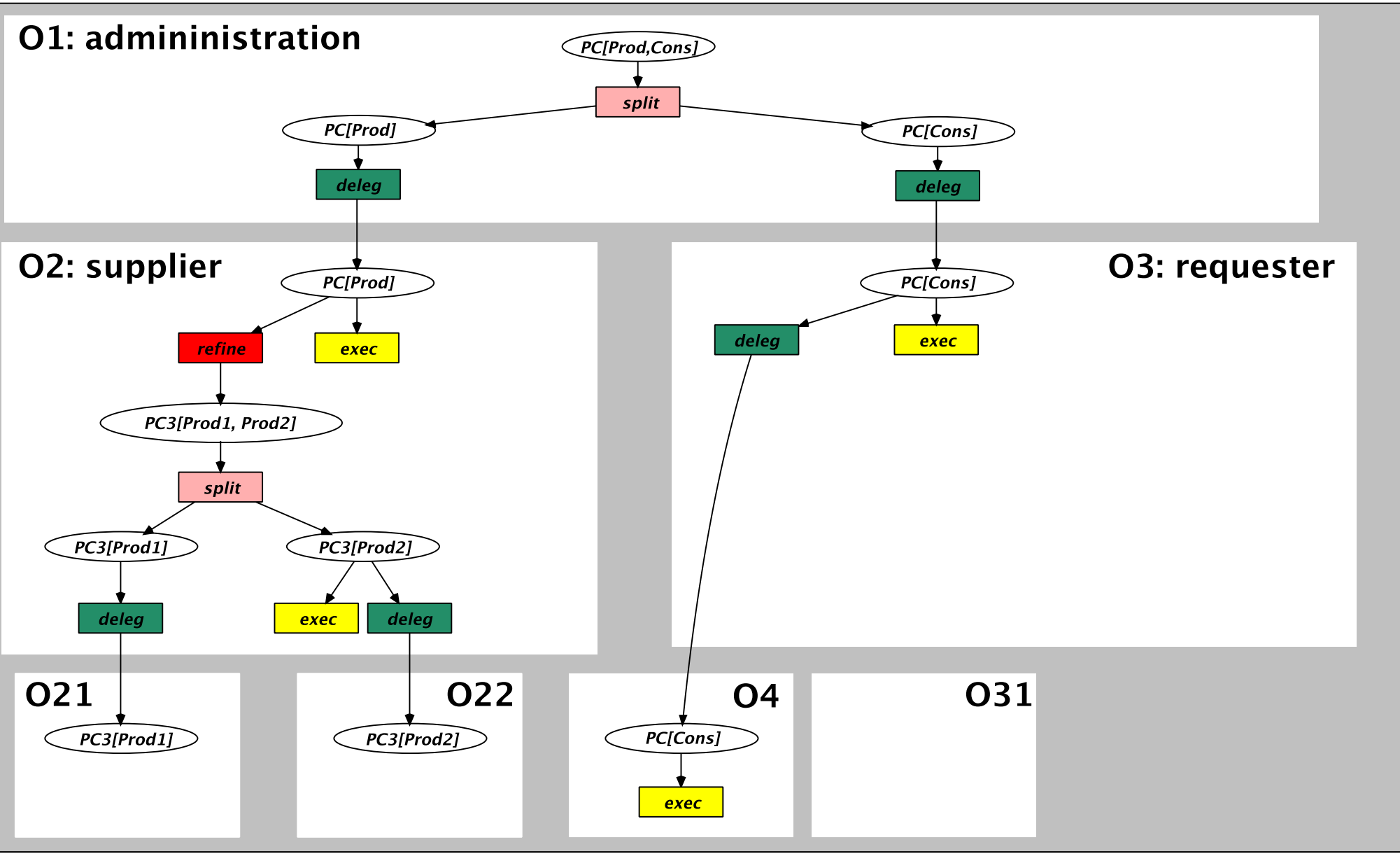
Transformations on SONAR-Models



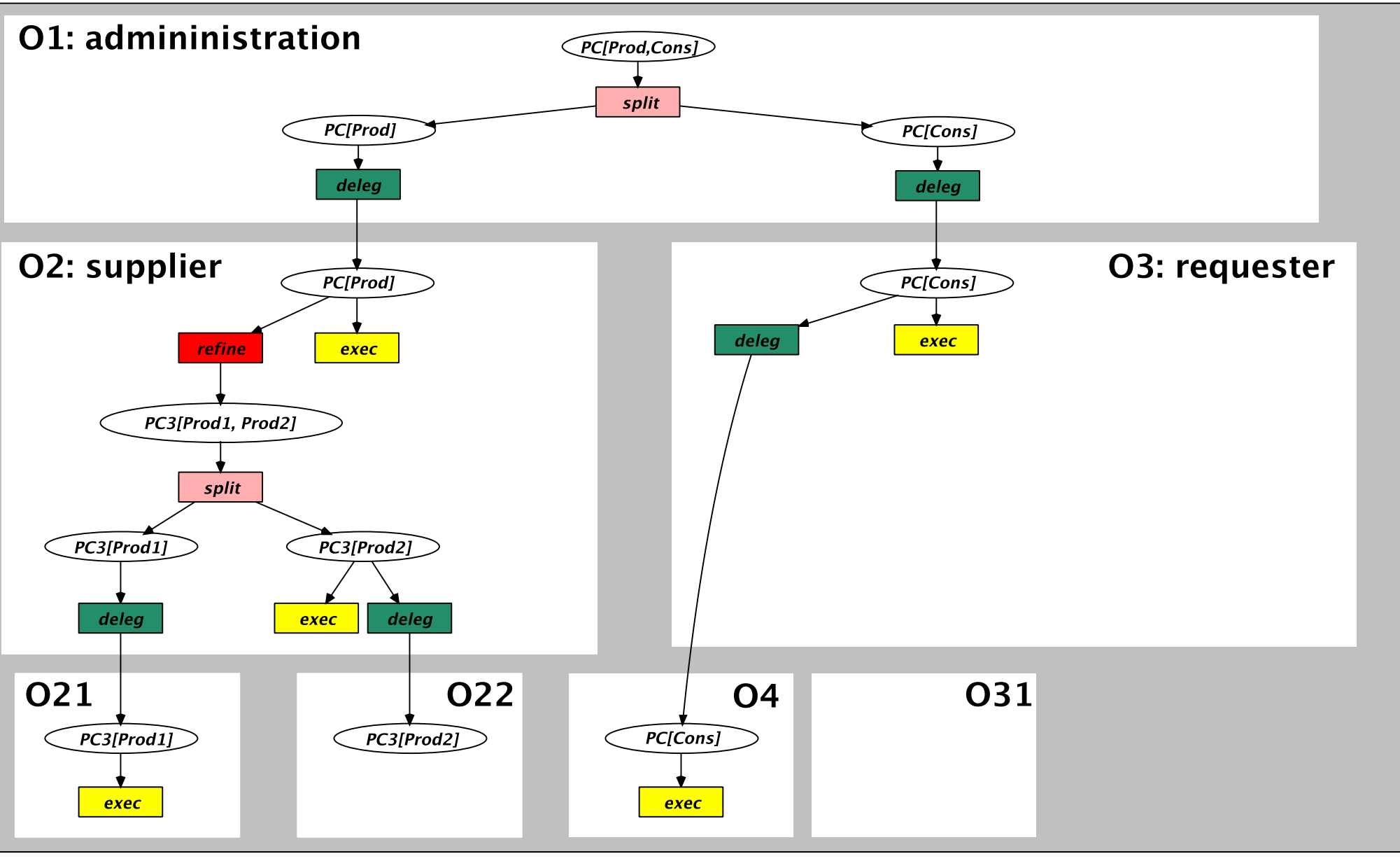
Transformations on SONAR-Models



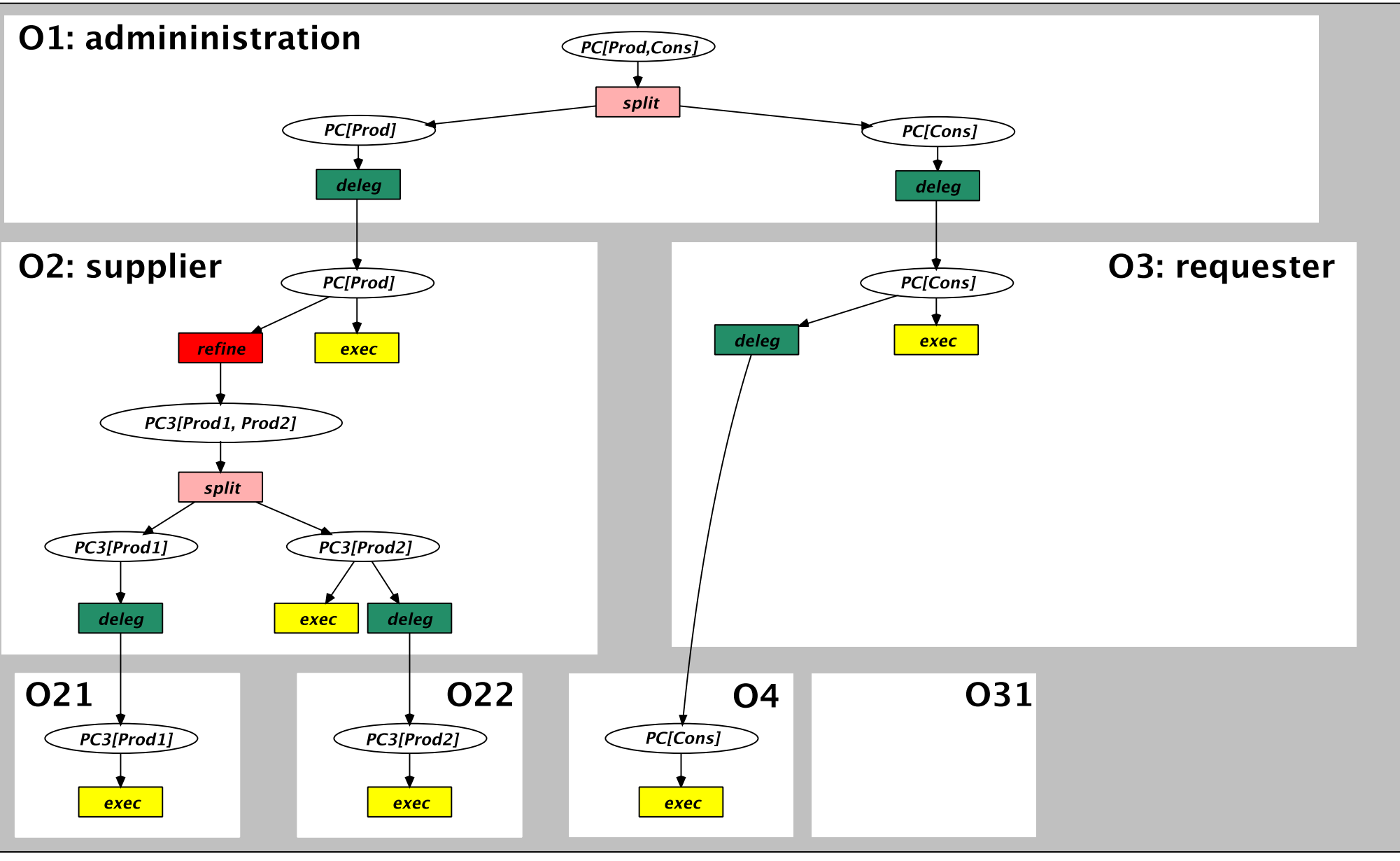
Transformations on SONAR-Models



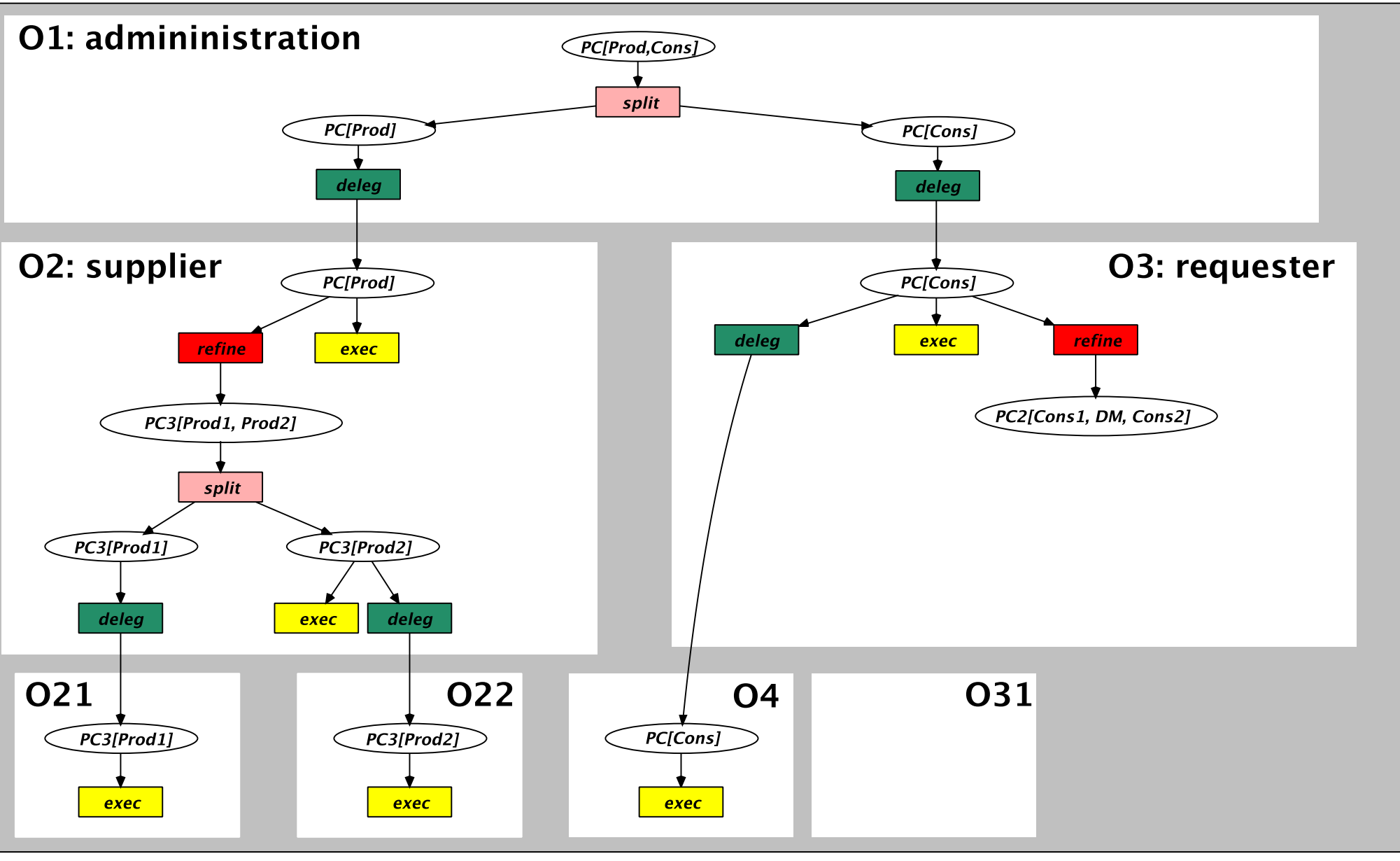
Transformations on SONAR-Models



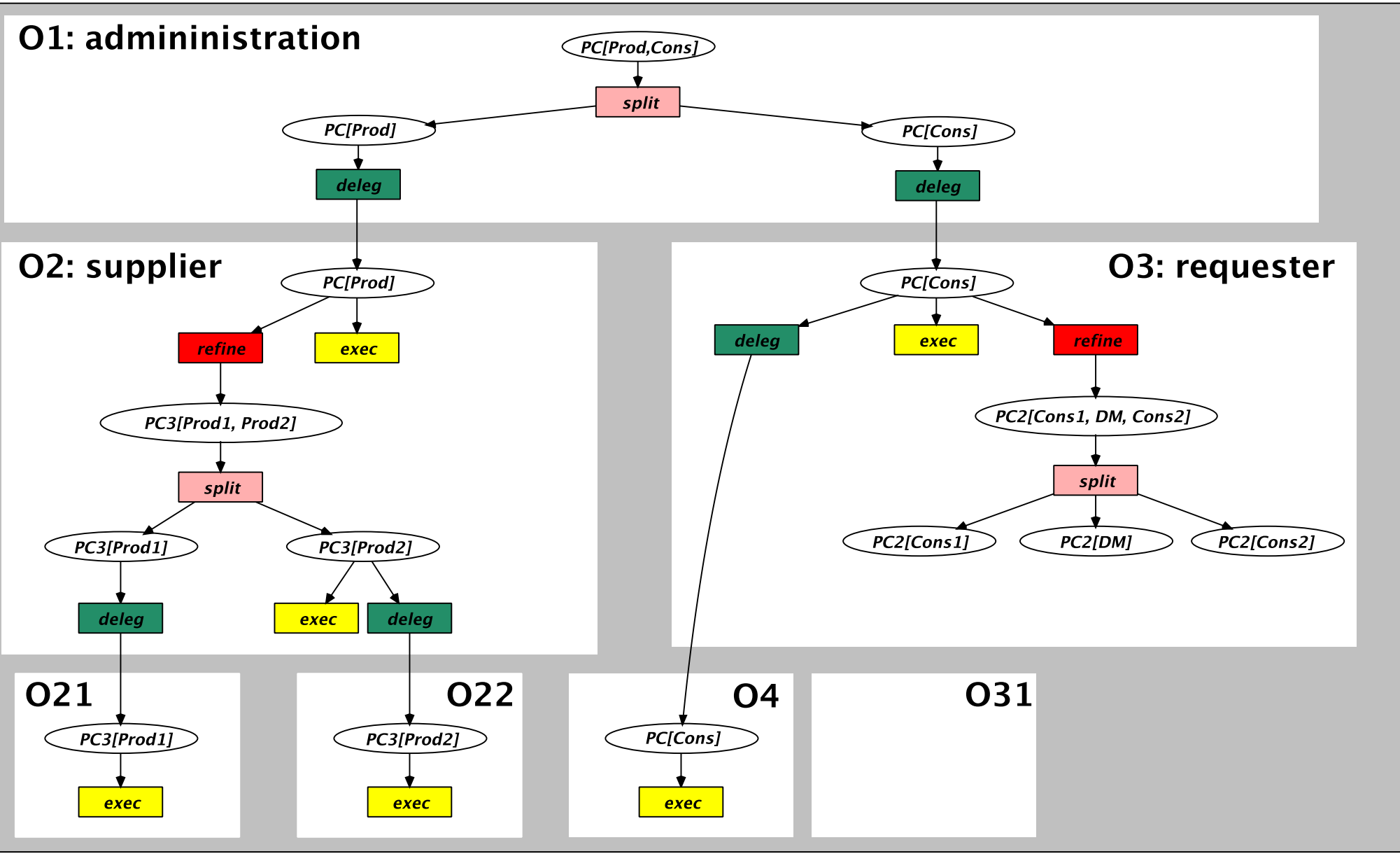
Transformations on SONAR-Models



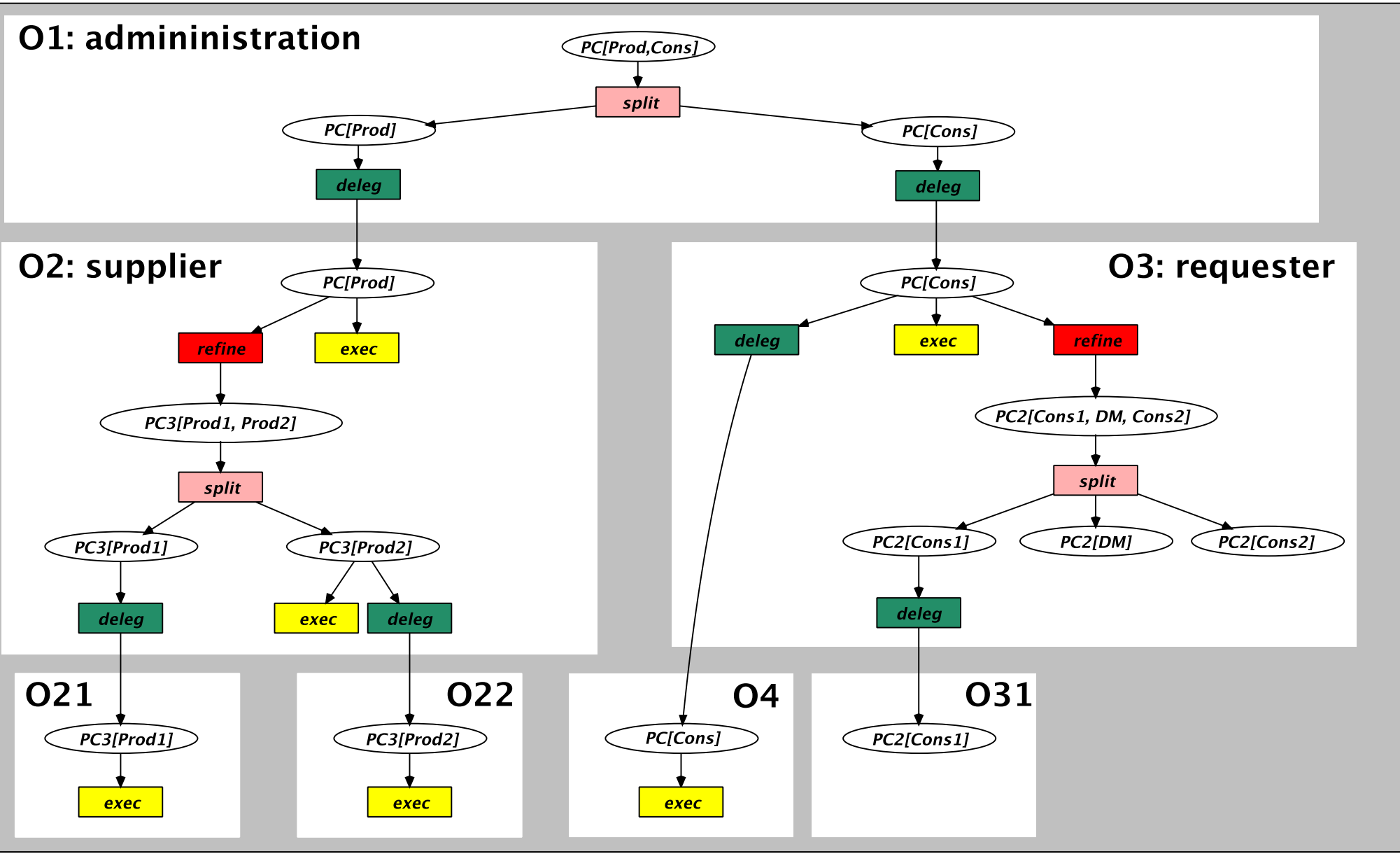
Transformations on SONAR-Models



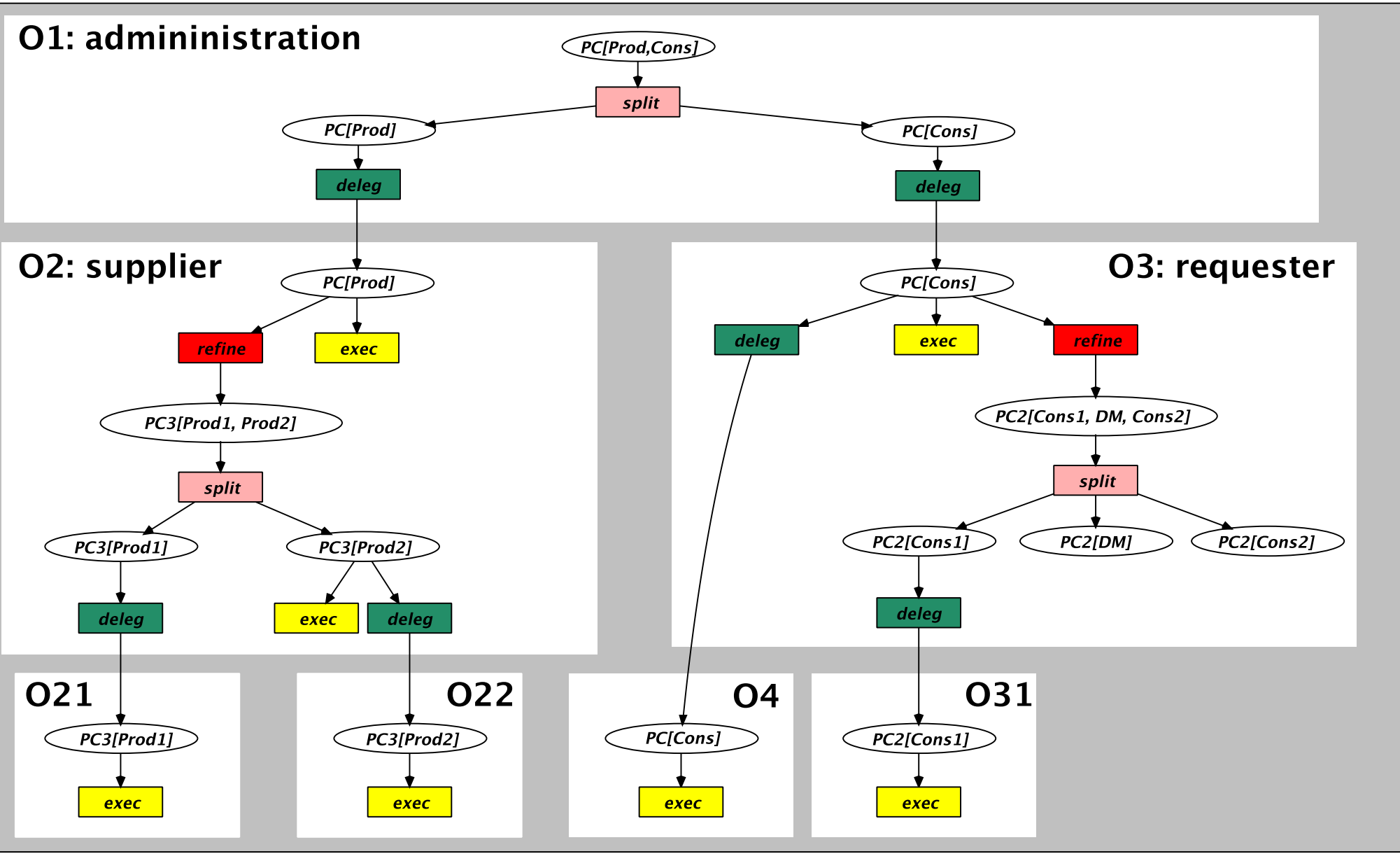
Transformations on SONAR-Models



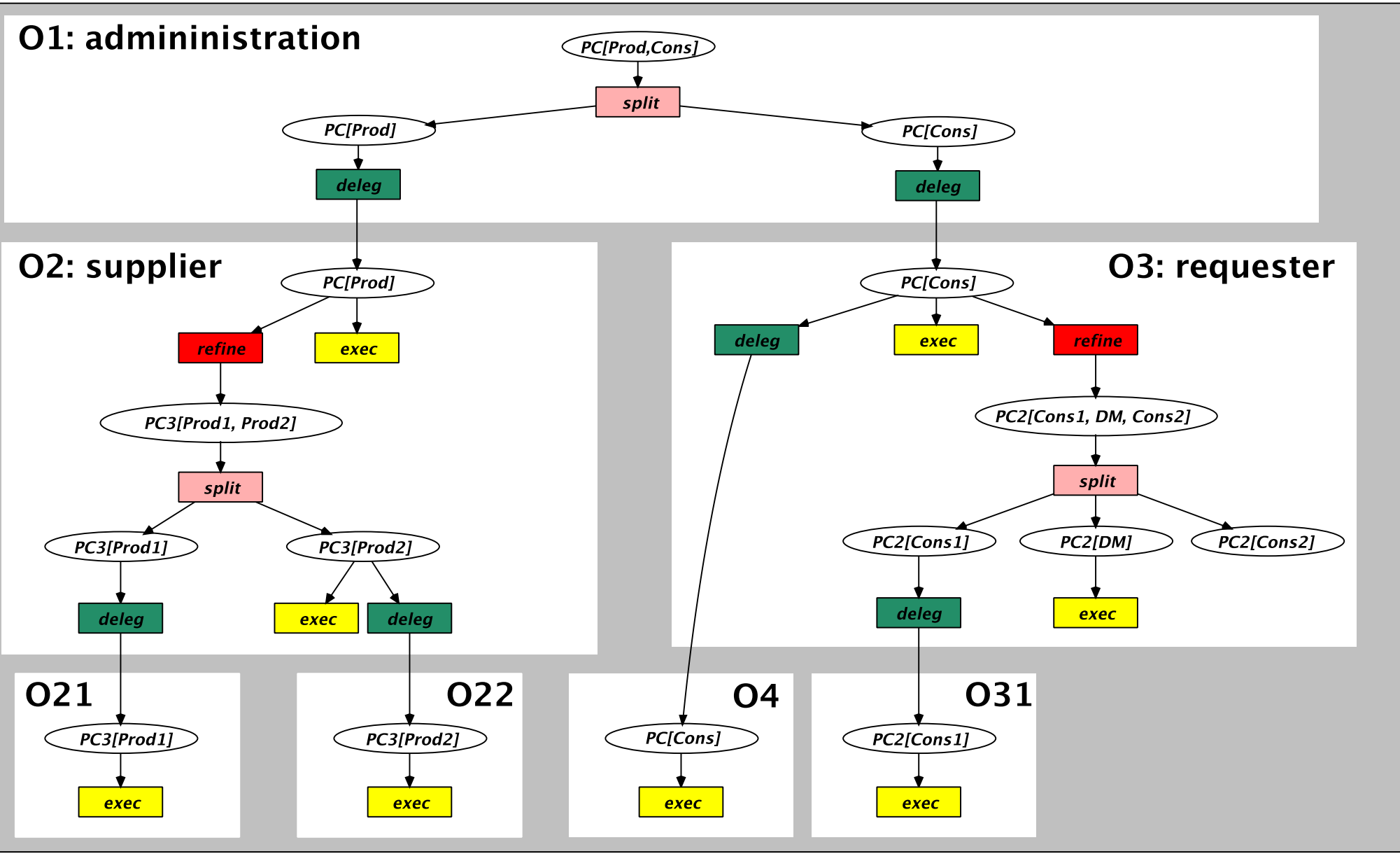
Transformations on SONAR-Models



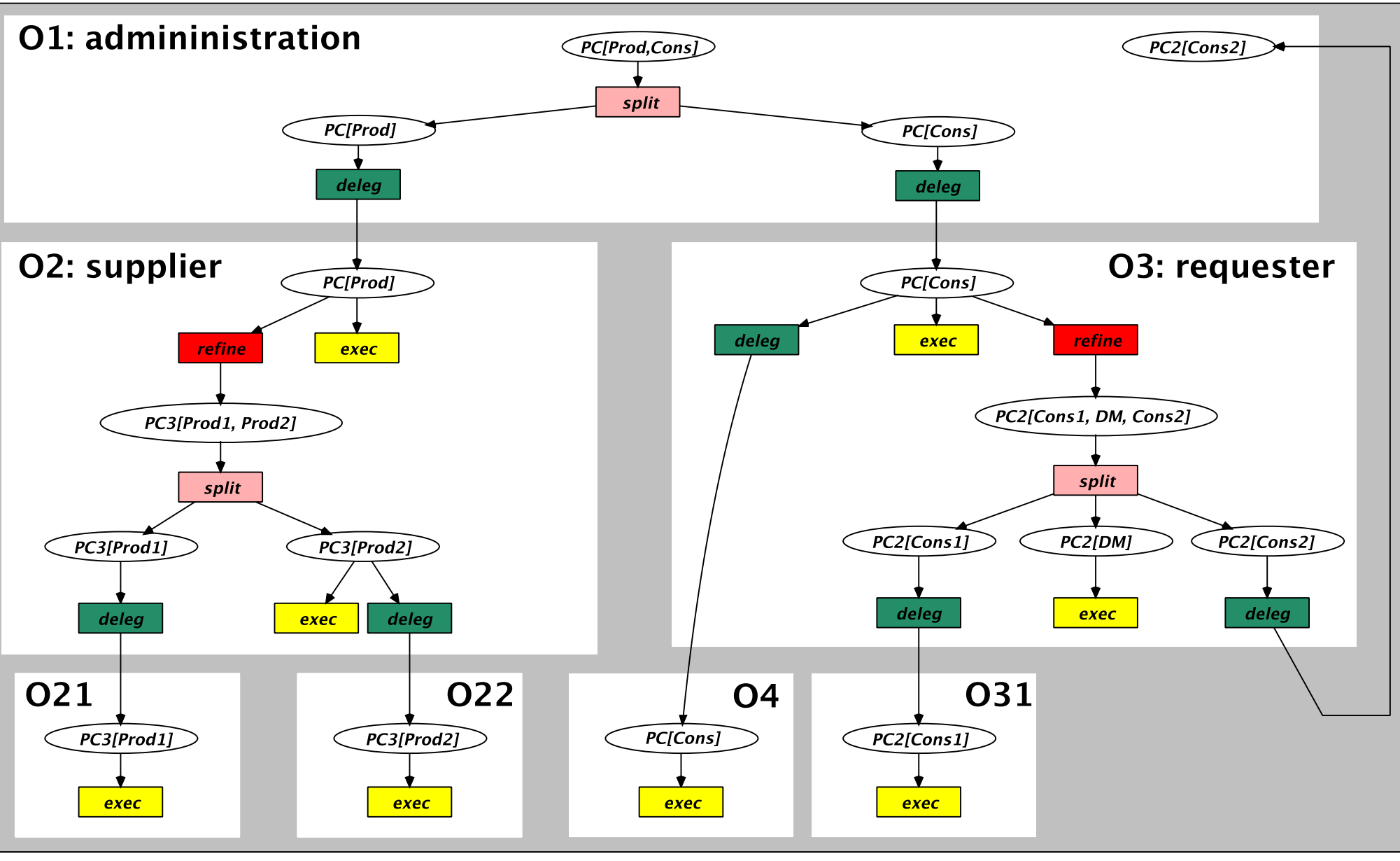
Transformations on SONAR-Models



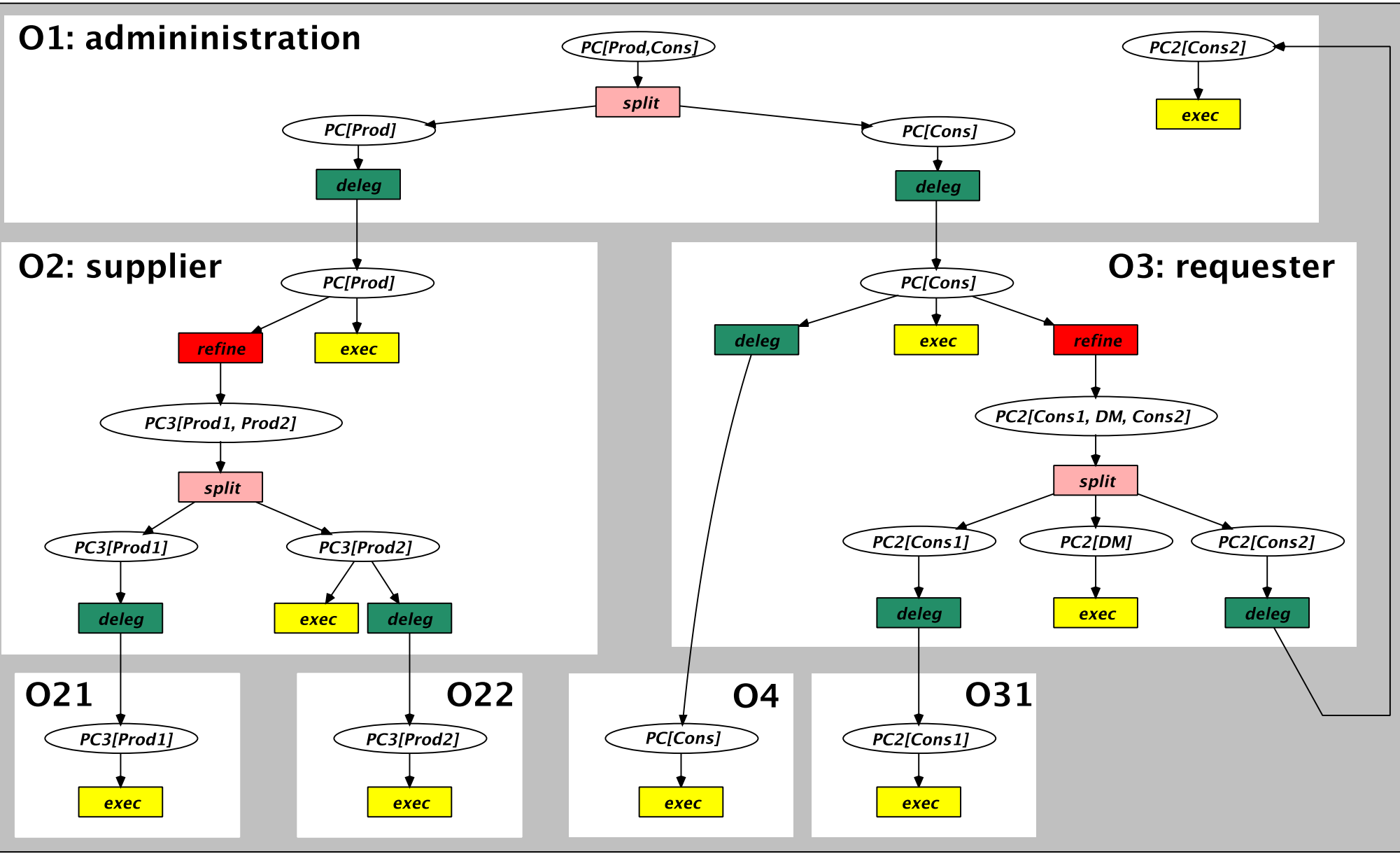
Transformations on SONAR-Models



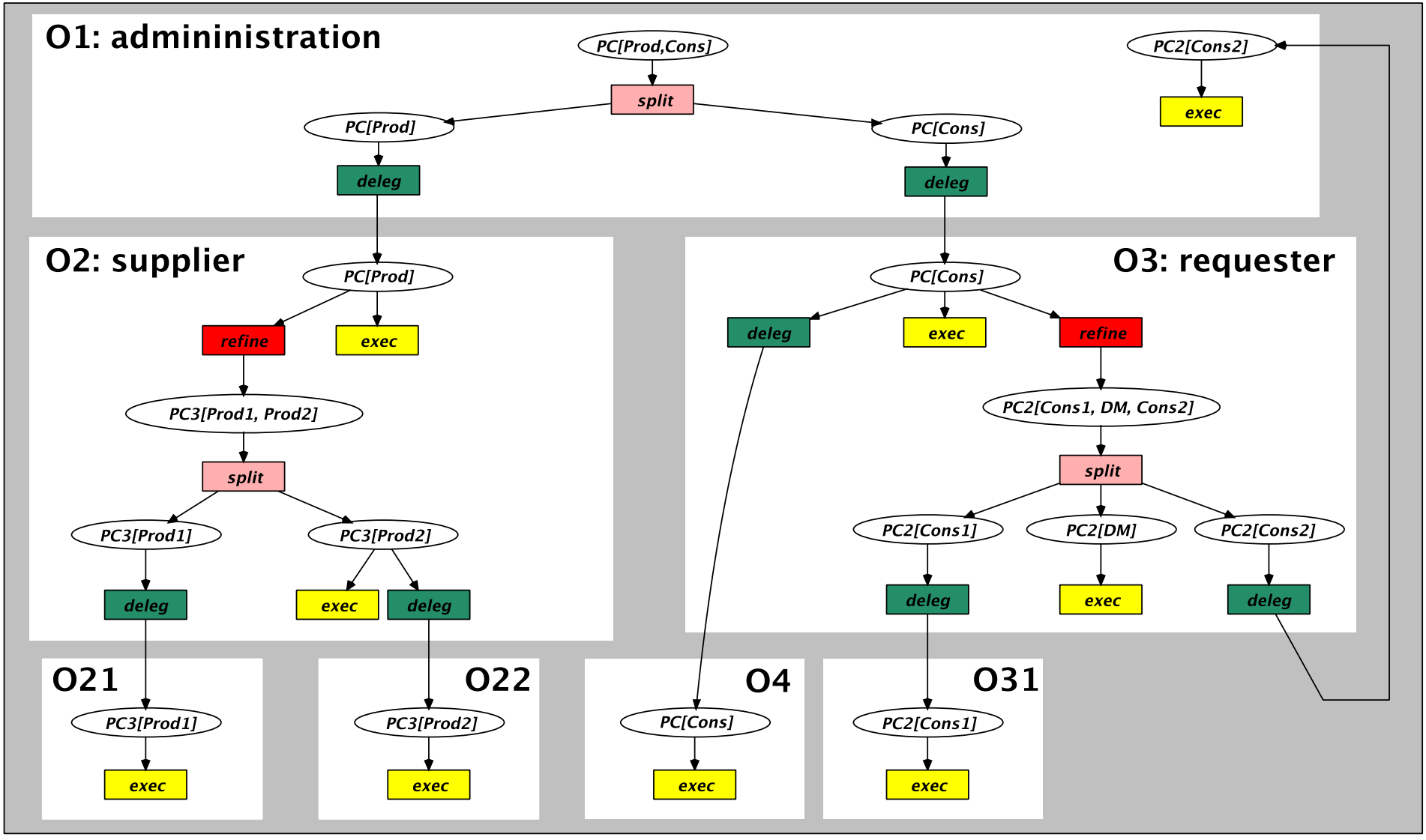
Transformations on SONAR-Models



Transformations on SONAR-Models



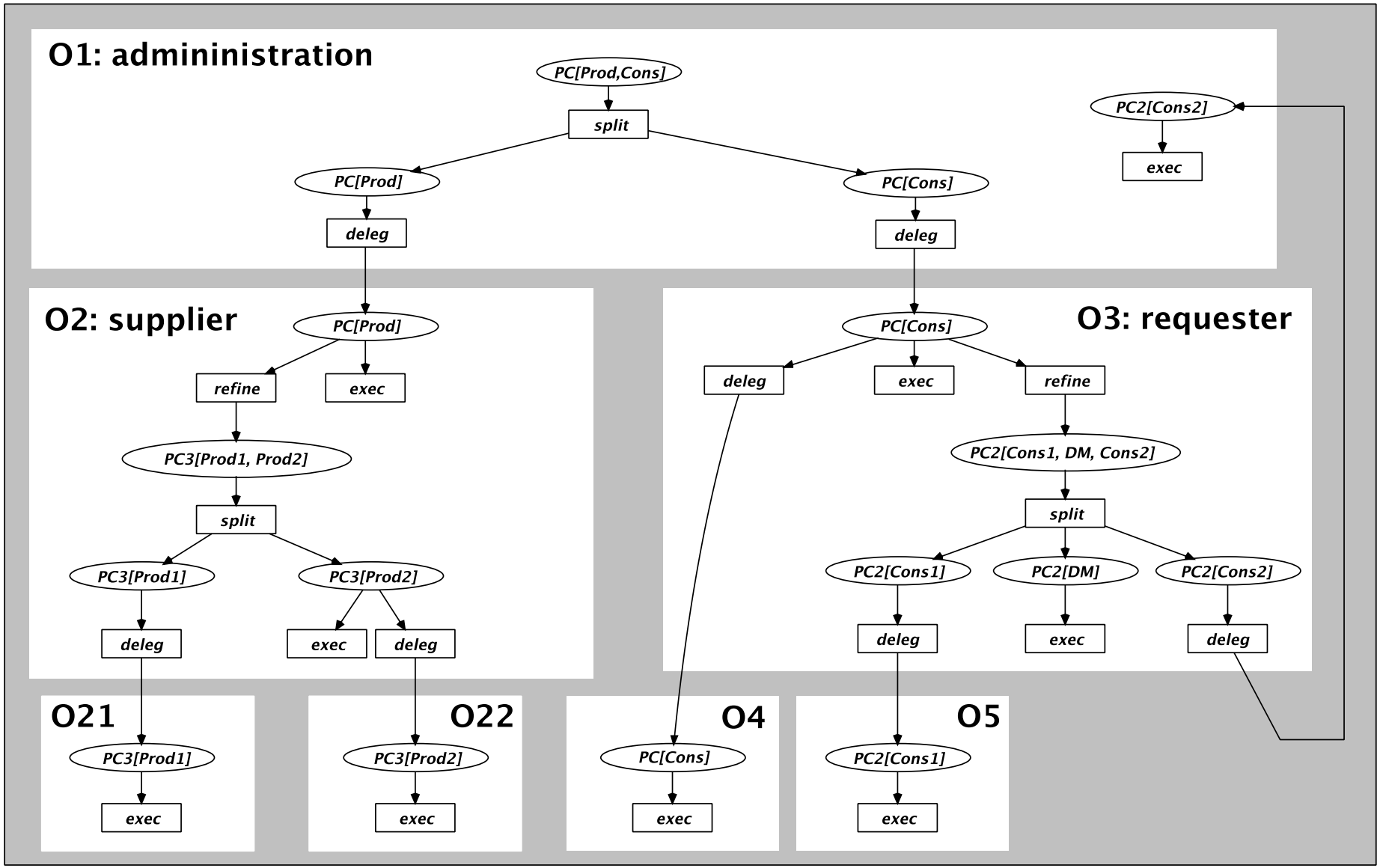
Example: SONAR-Model



Outline

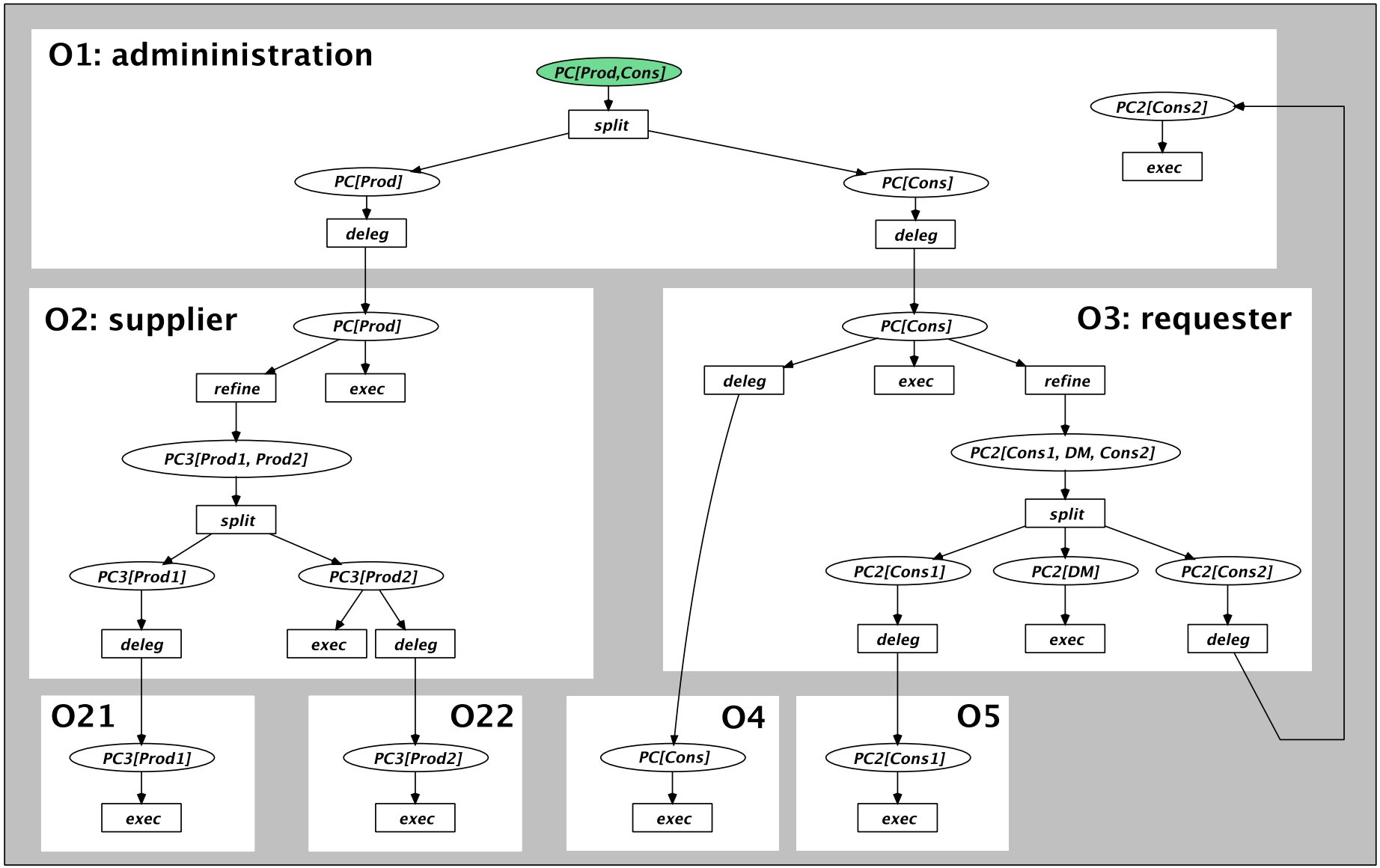
Teamformation in SONAR

Team = Petri Net Process of the Organisation Net



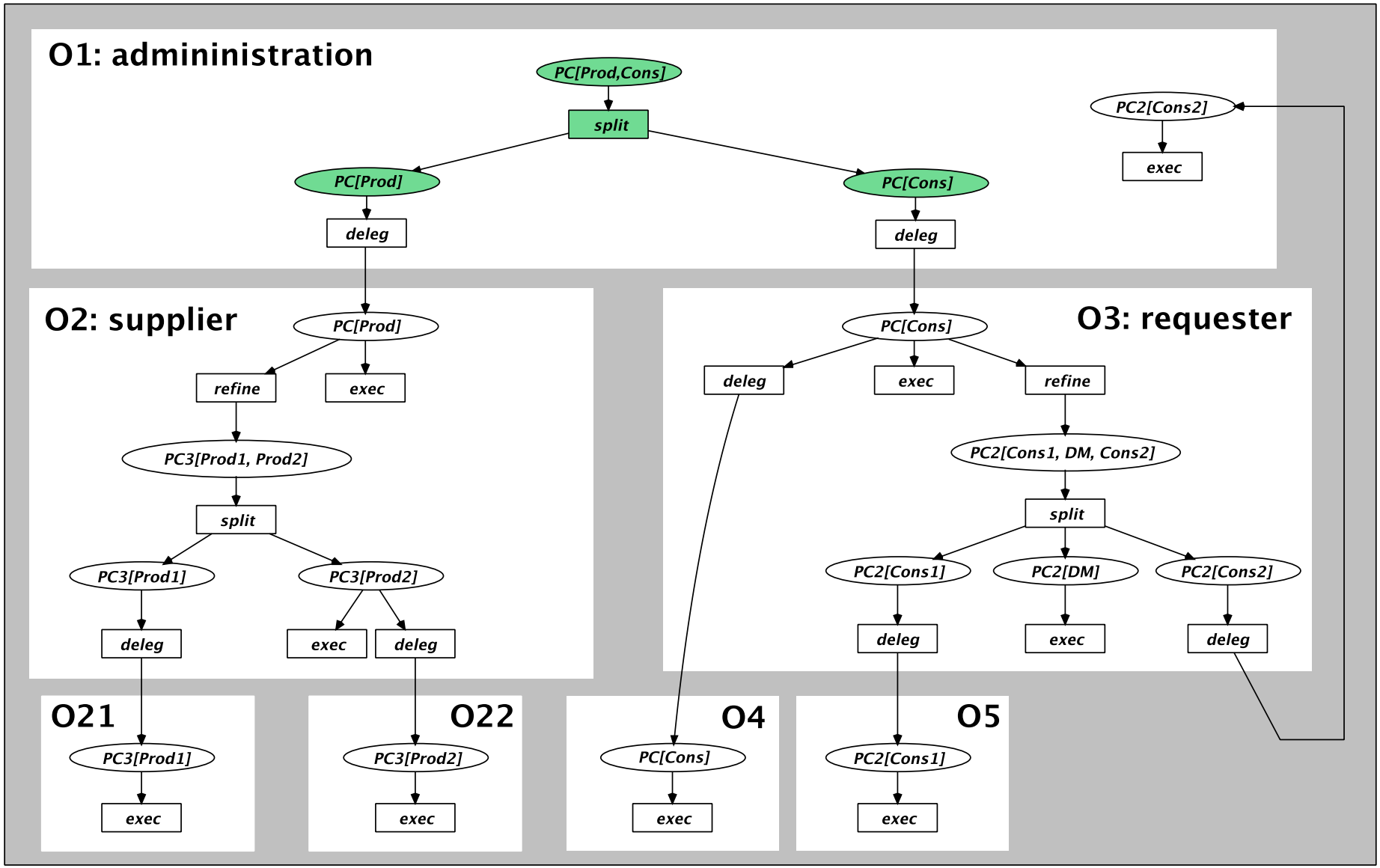
Teamformation in SONAR

Team = Petri Net Process of the Organisation Net



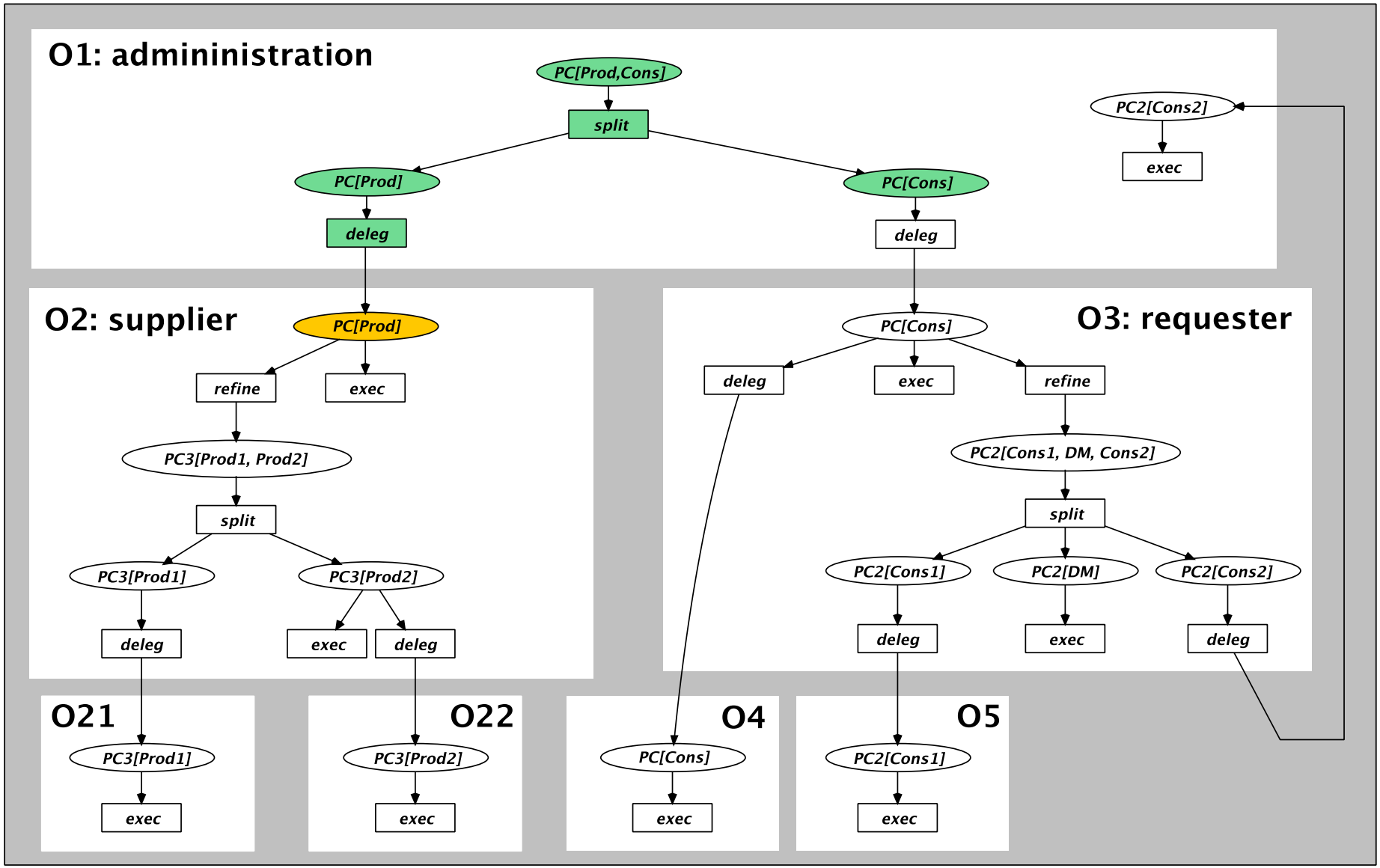
Teamformation in SONAR

Team = Petri Net Process of the Organisation Net



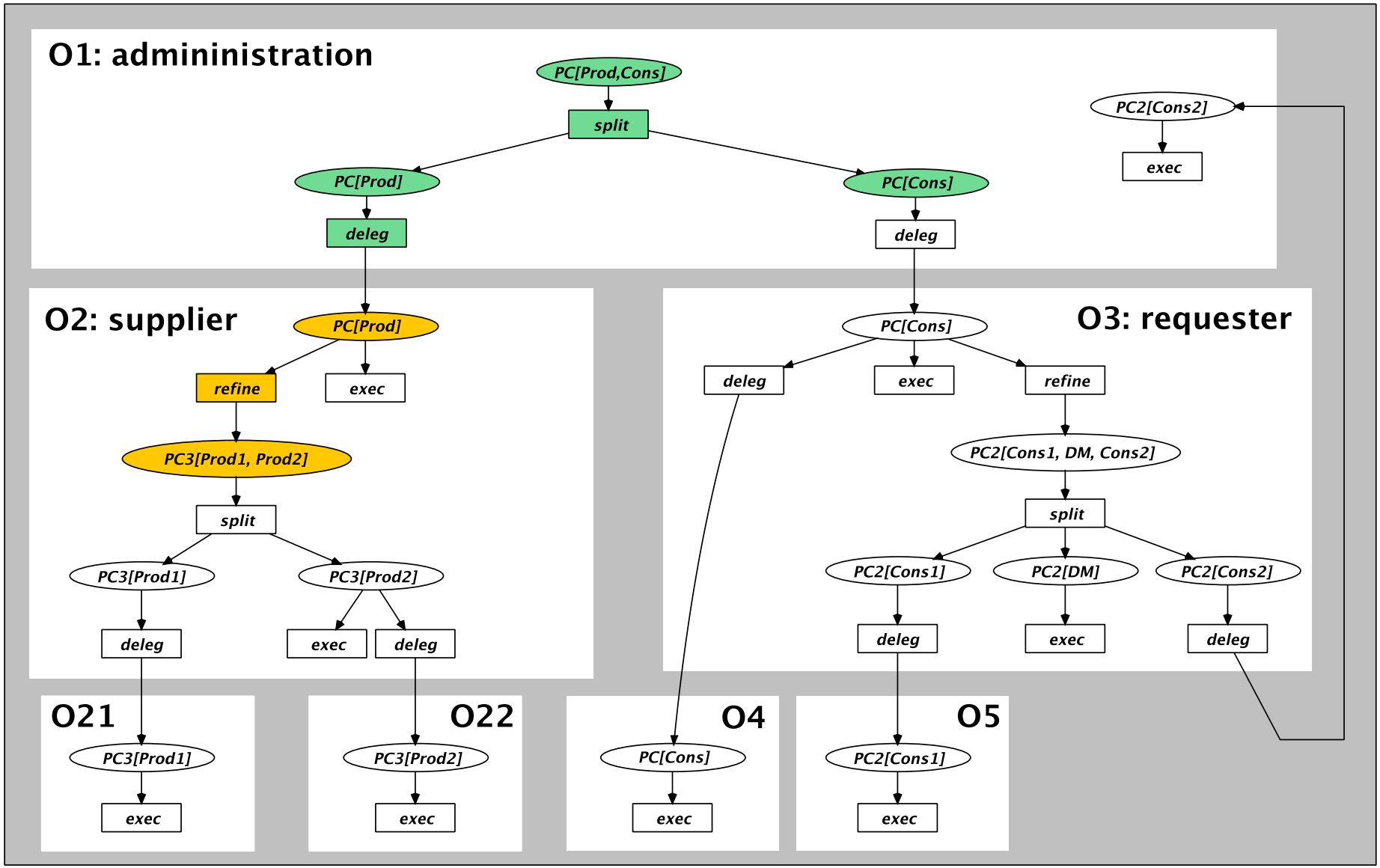
Teamformation in SONAR

Team = Petri Net Process of the Organisation Net



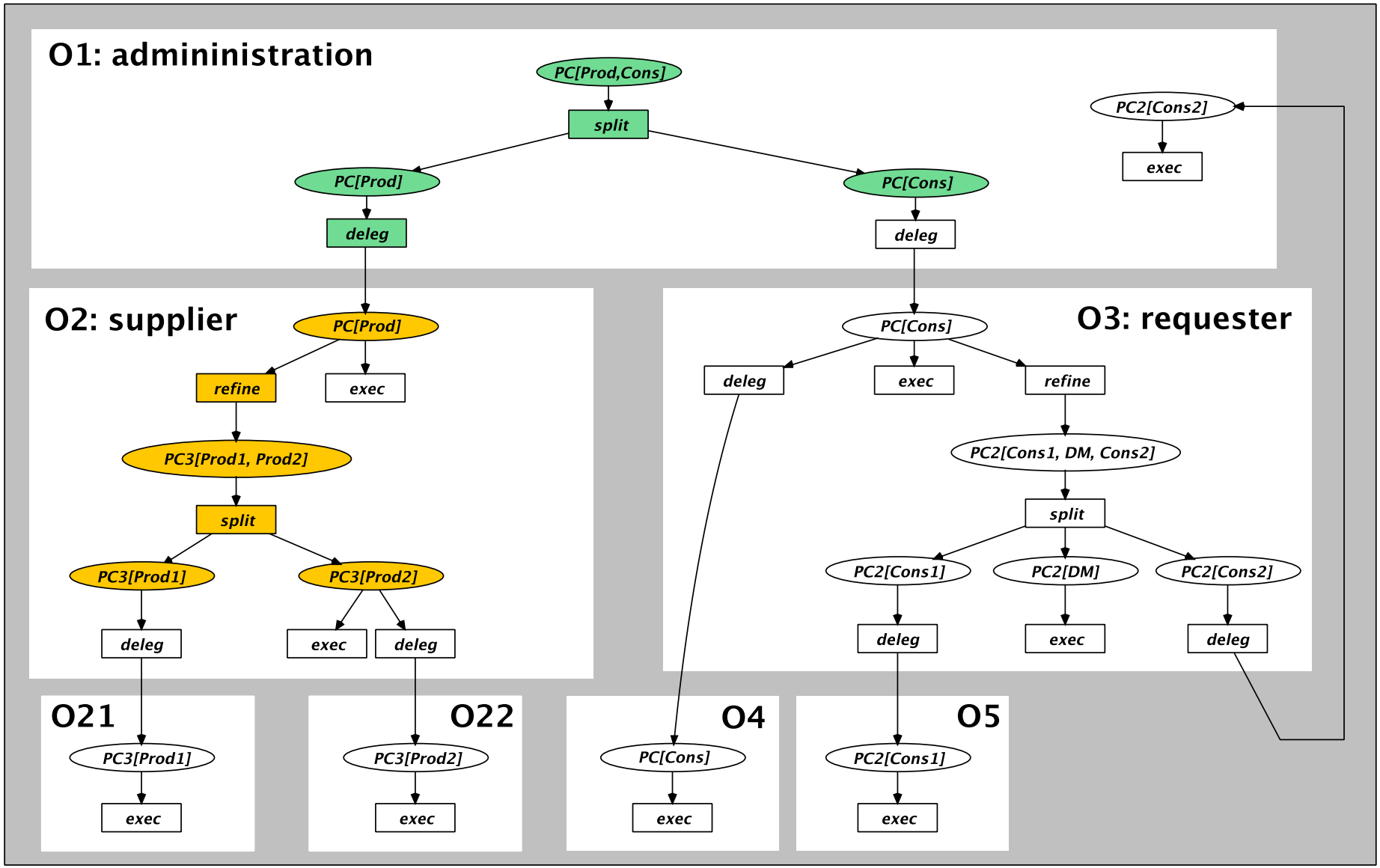
Teamformation in SONAR

Team = Petri Net Process of the Organisation Net



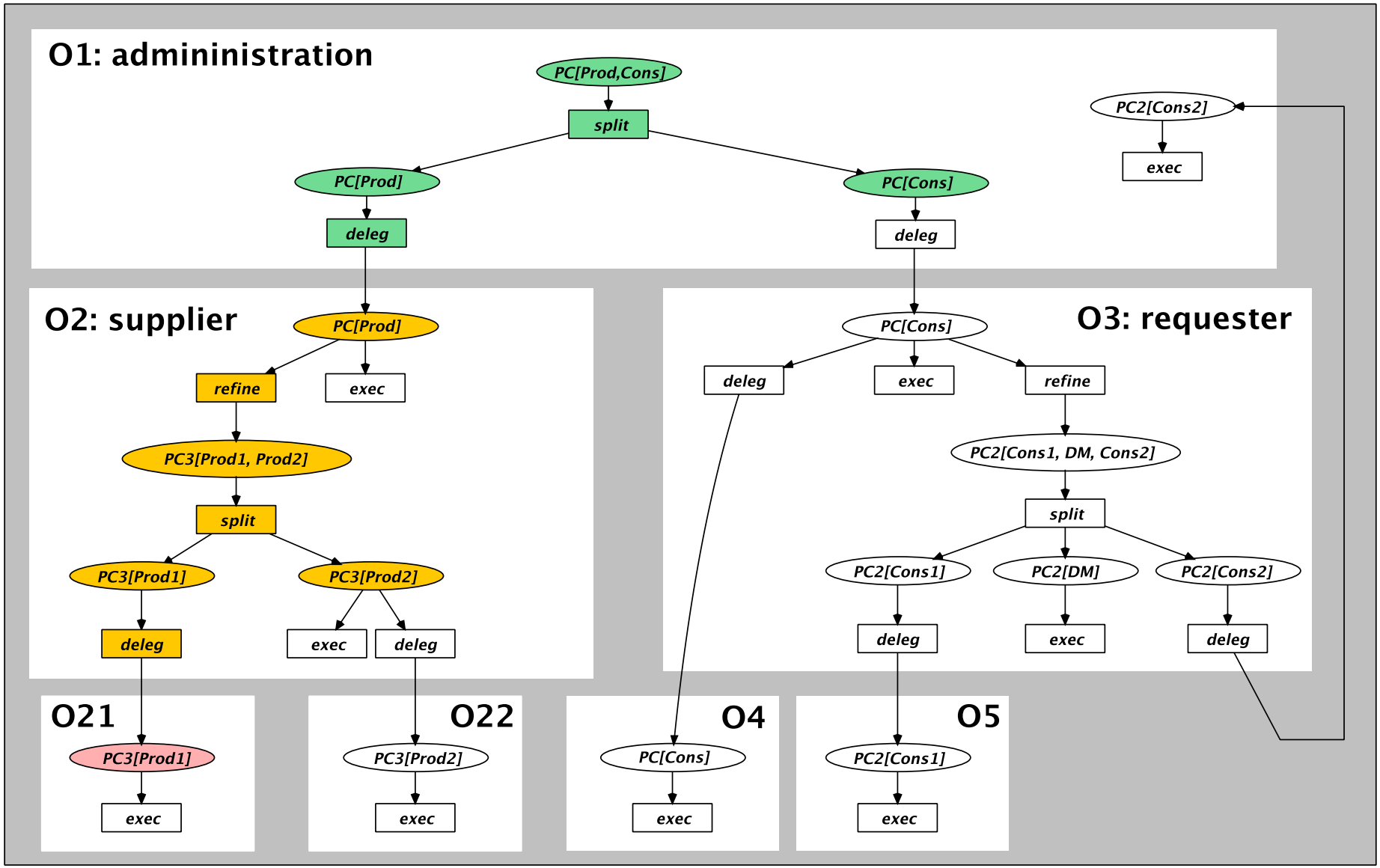
Teamformation in SONAR

Team = Petri Net Process of the Organisation Net



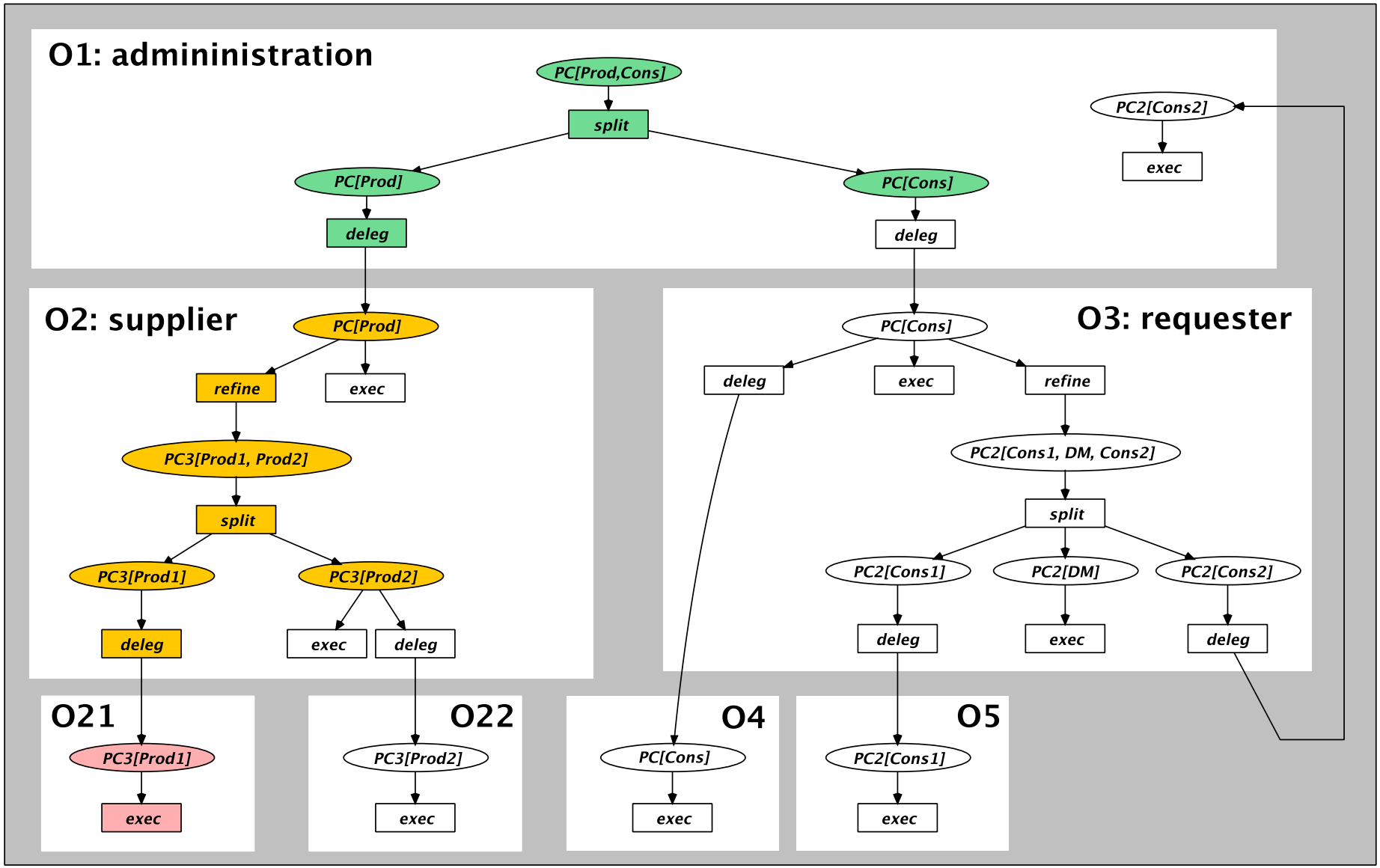
Teamformation in SONAR

Team = Petri Net Process of the Organisation Net



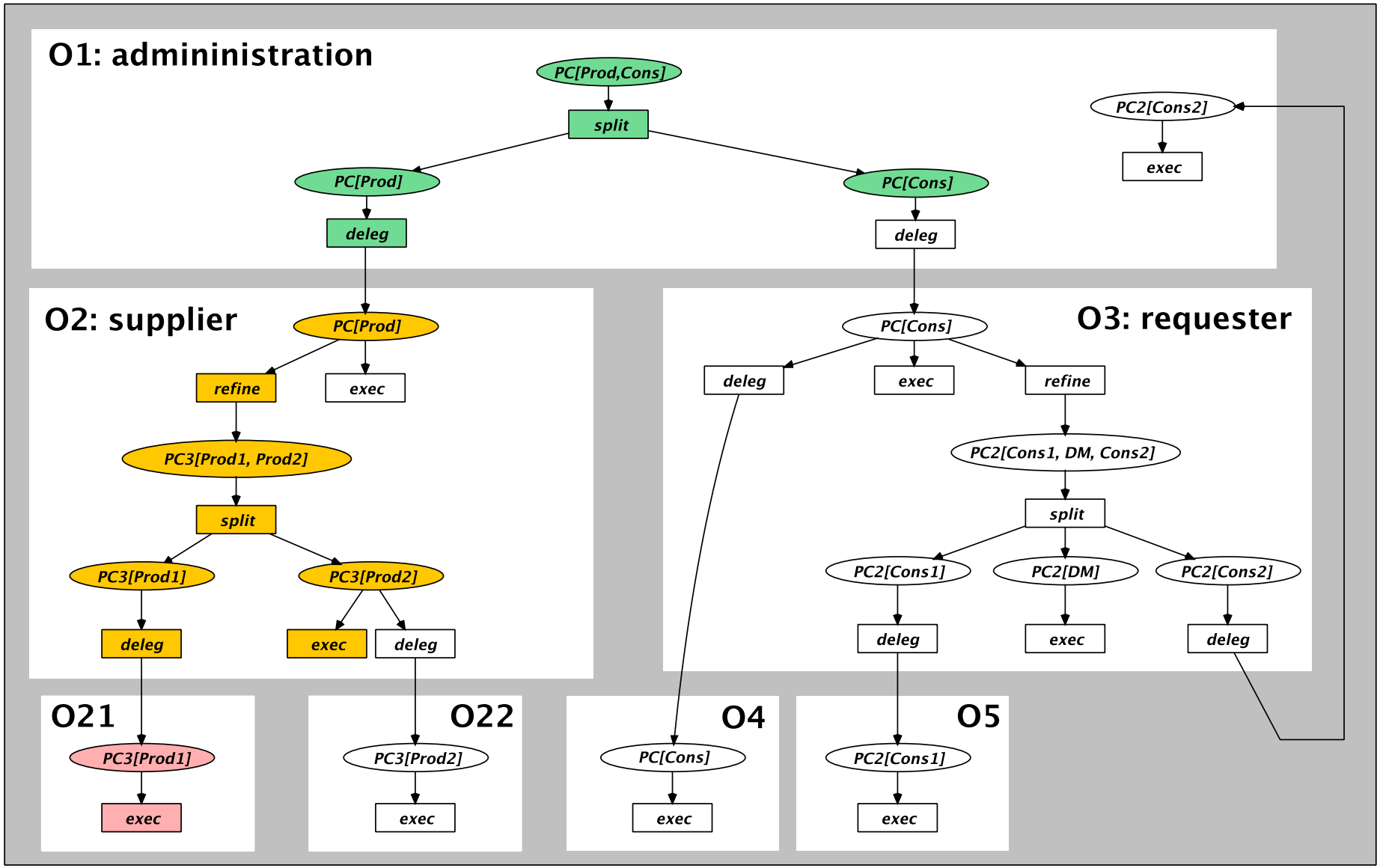
Teamformation in SONAR

Team = Petri Net Process of the Organisation Net



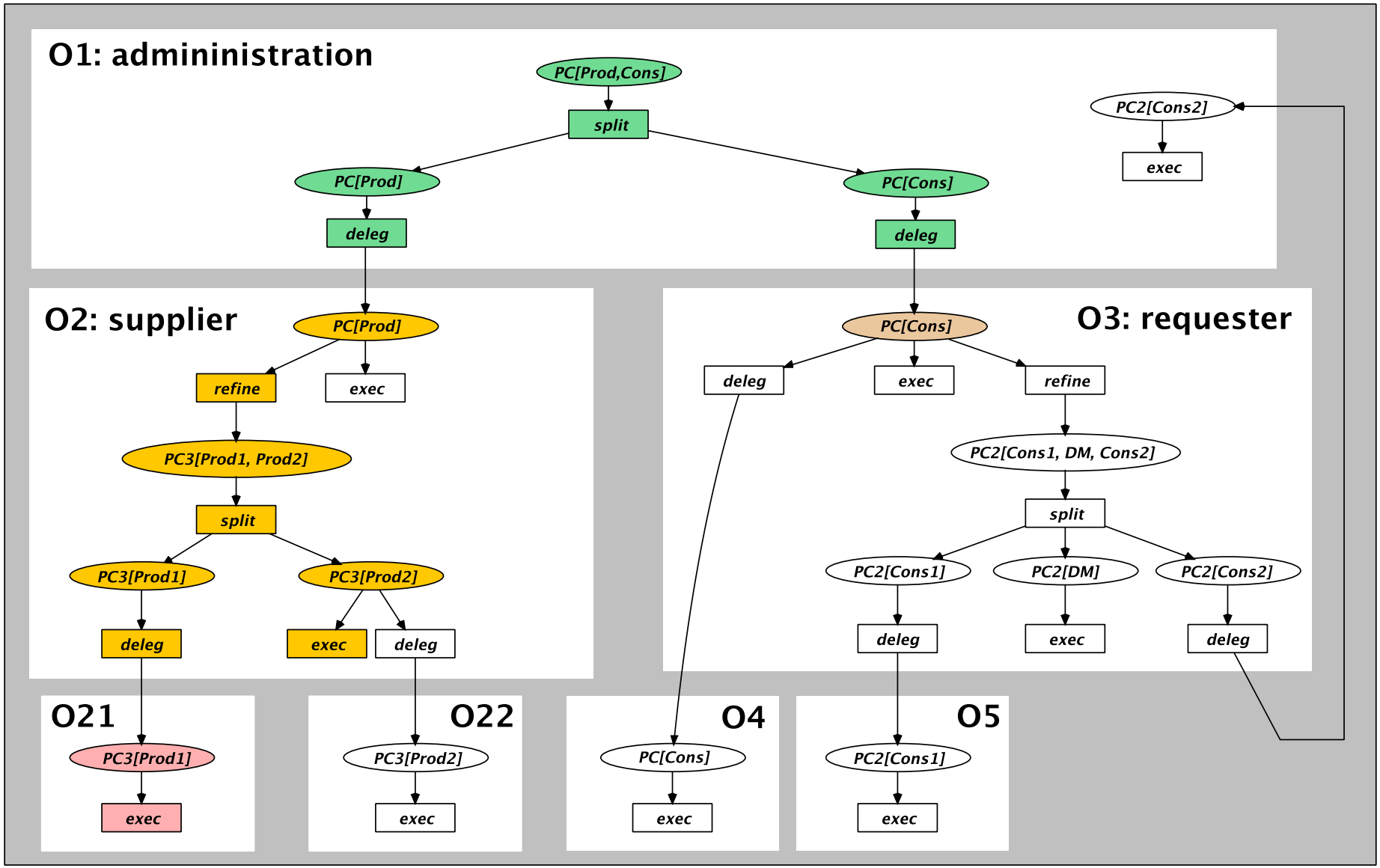
Teamformation in SONAR

Team = Petri Net Process of the Organisation Net



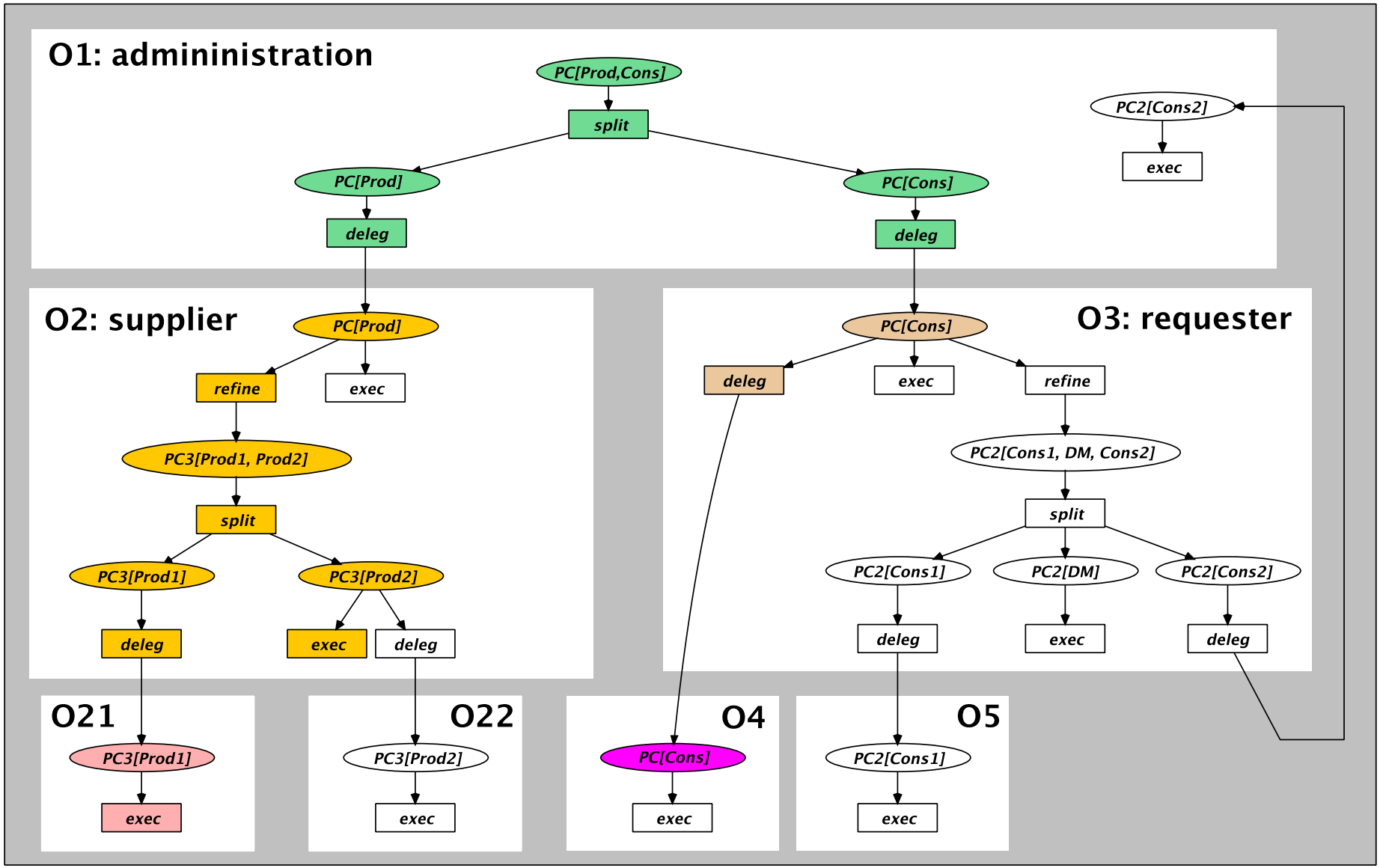
Teamformation in SONAR

Team = Petri Net Process of the Organisation Net



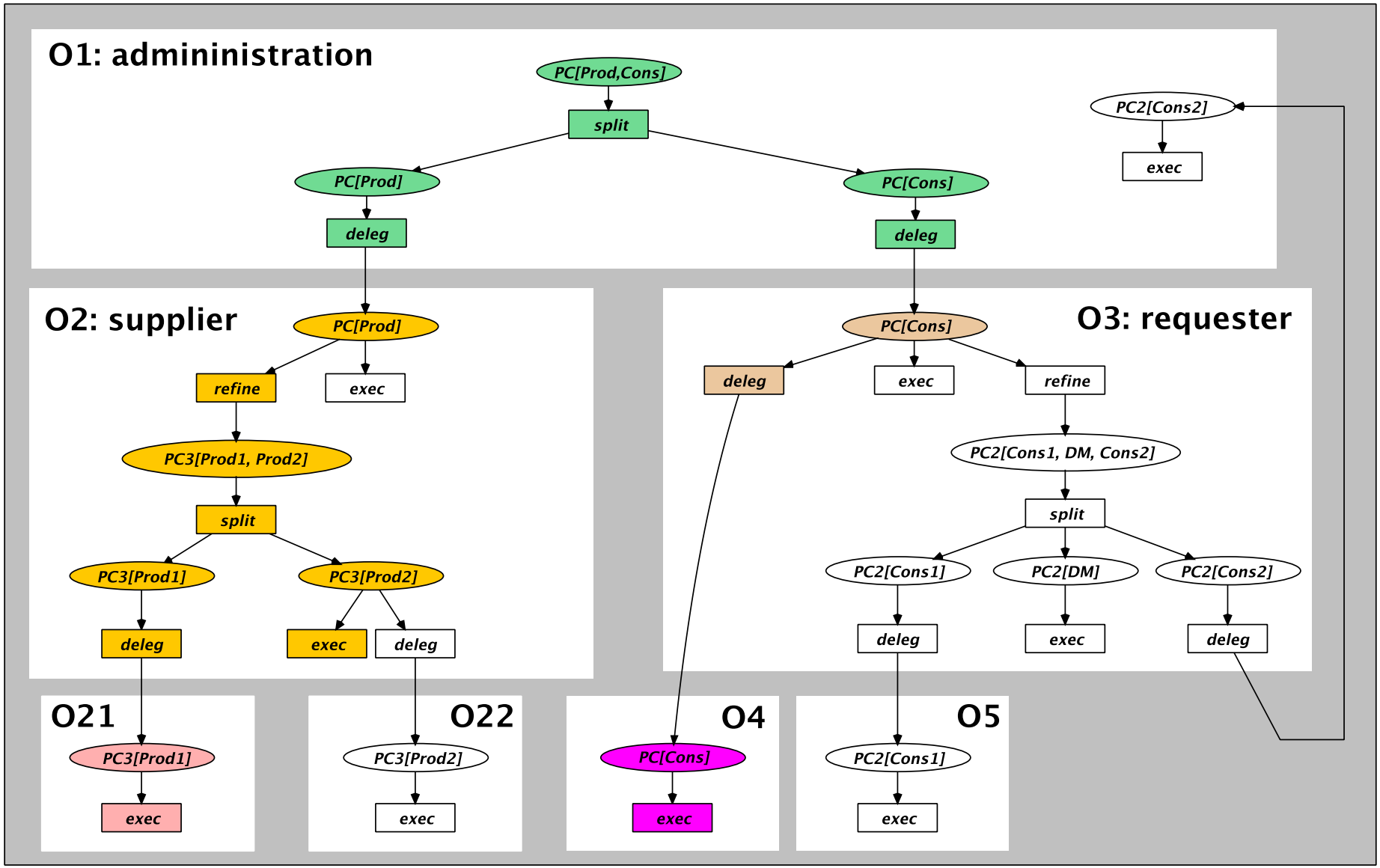
Teamformation in SONAR

Team = Petri Net Process of the Organisation Net



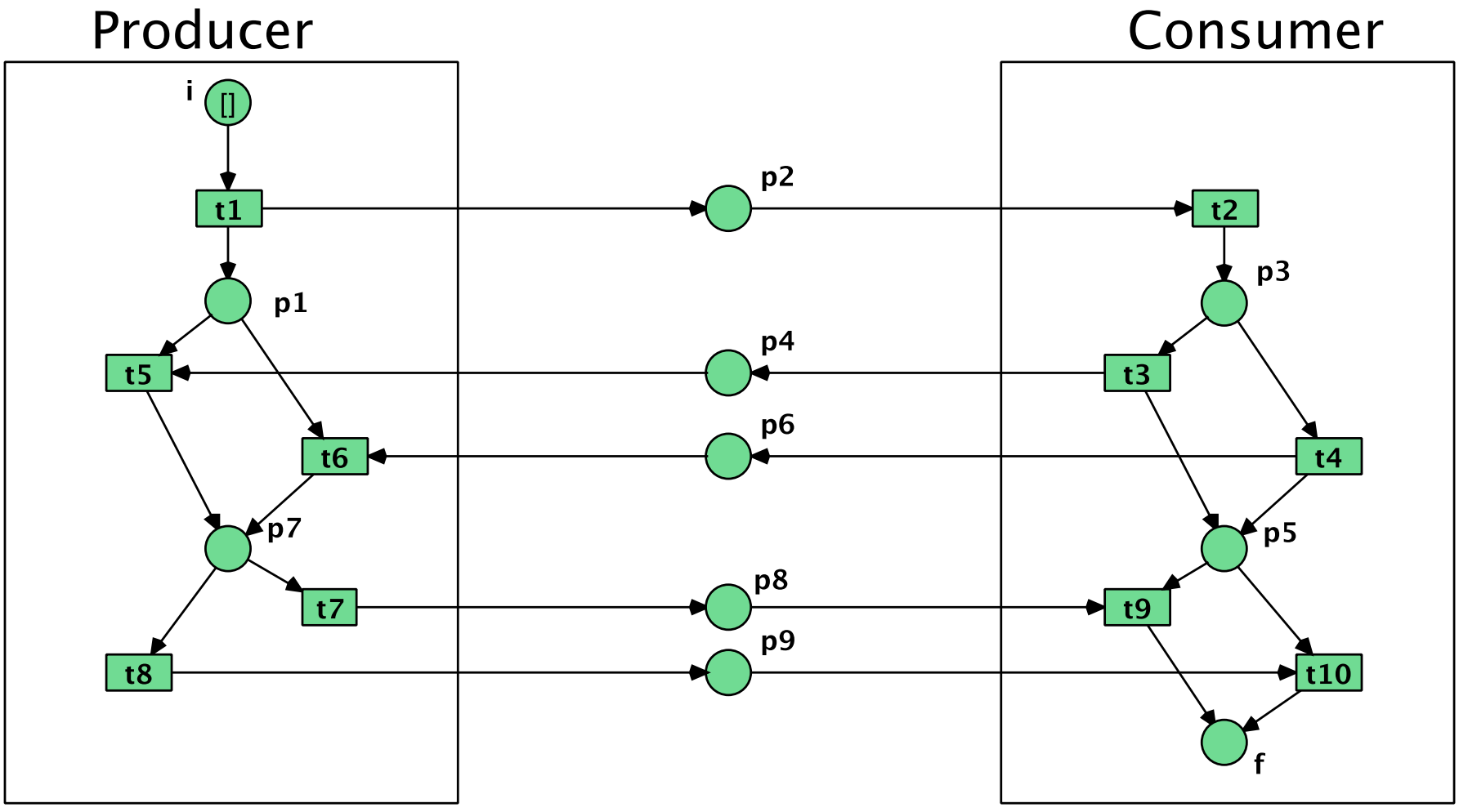
Teamformation in SONAR

Team = Petri Net Process of the Organisation Net



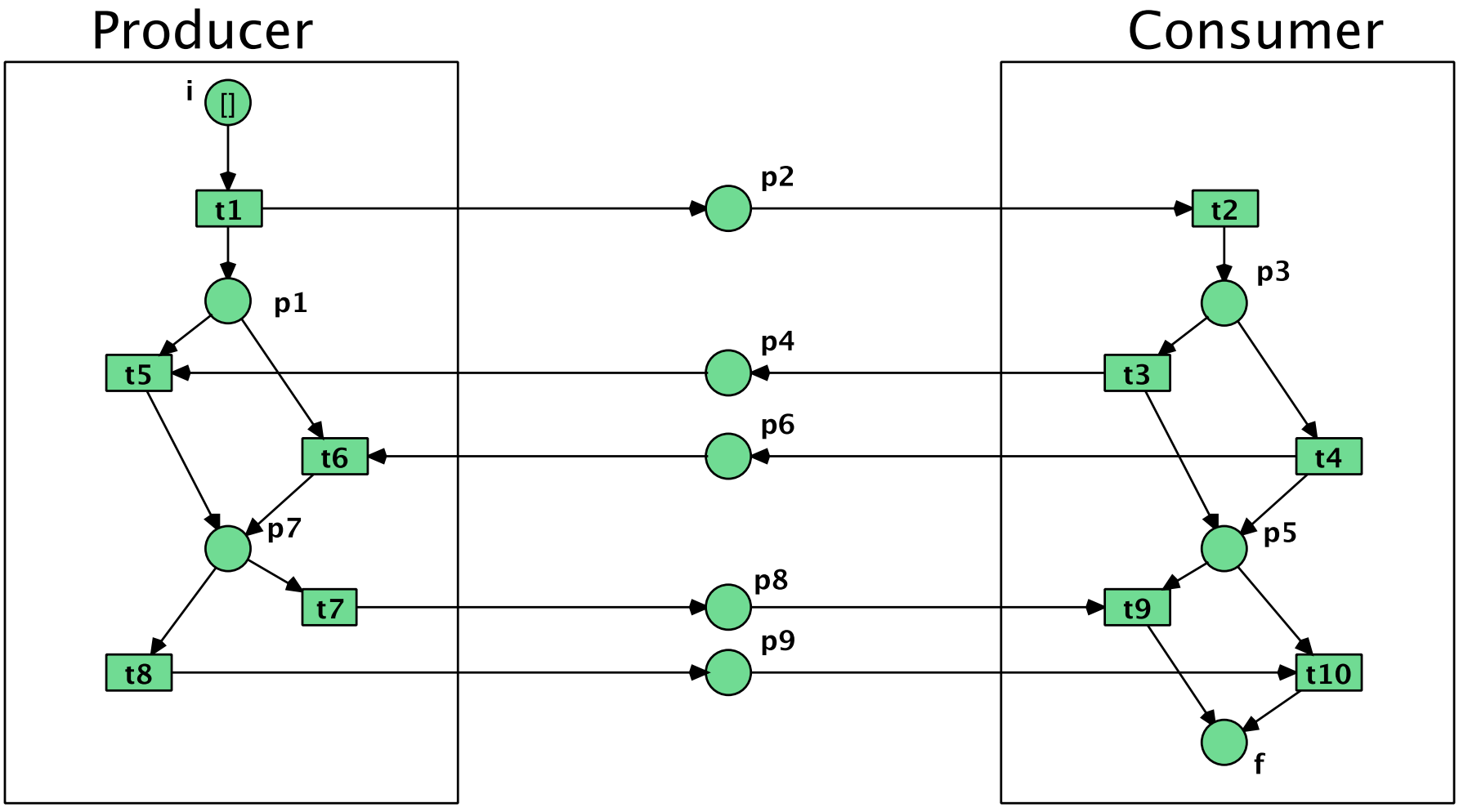
Outline

The Team's Interaction Protocol



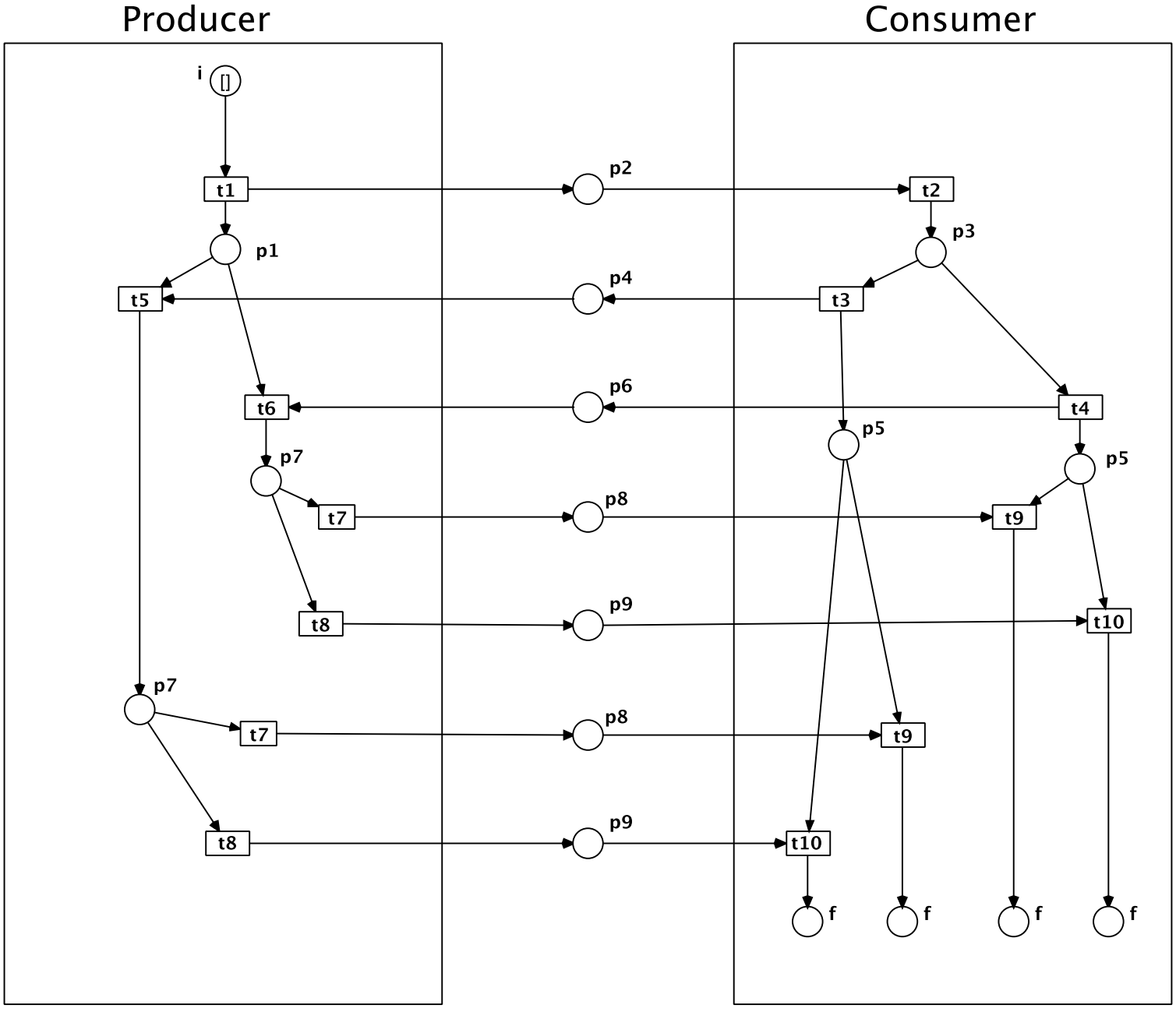
Compute the **Unfolding** of the Interaction Protocol.

The Team's Interaction Protocol

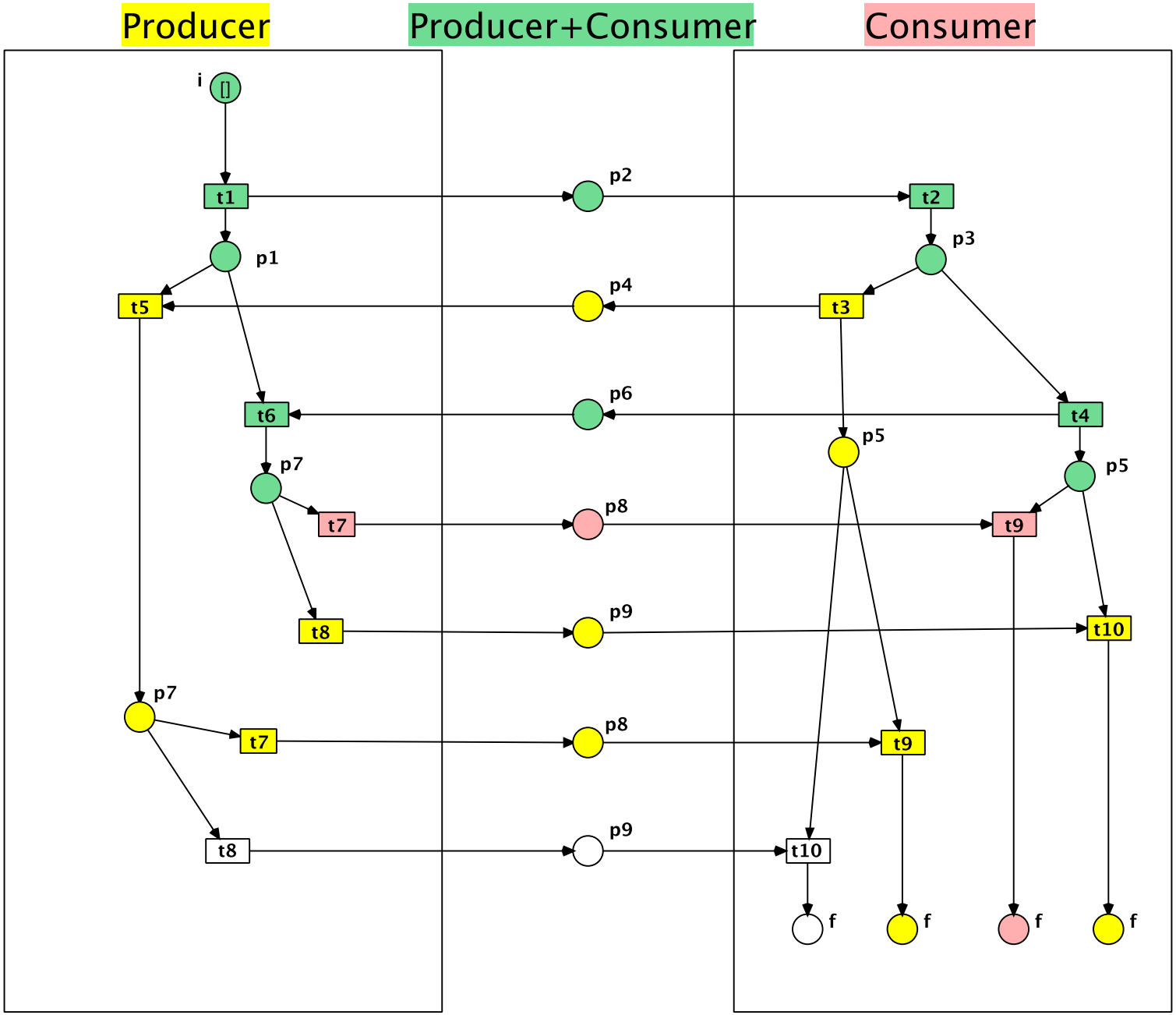


Compute the **Unfolding** of the Interaction Protocol.

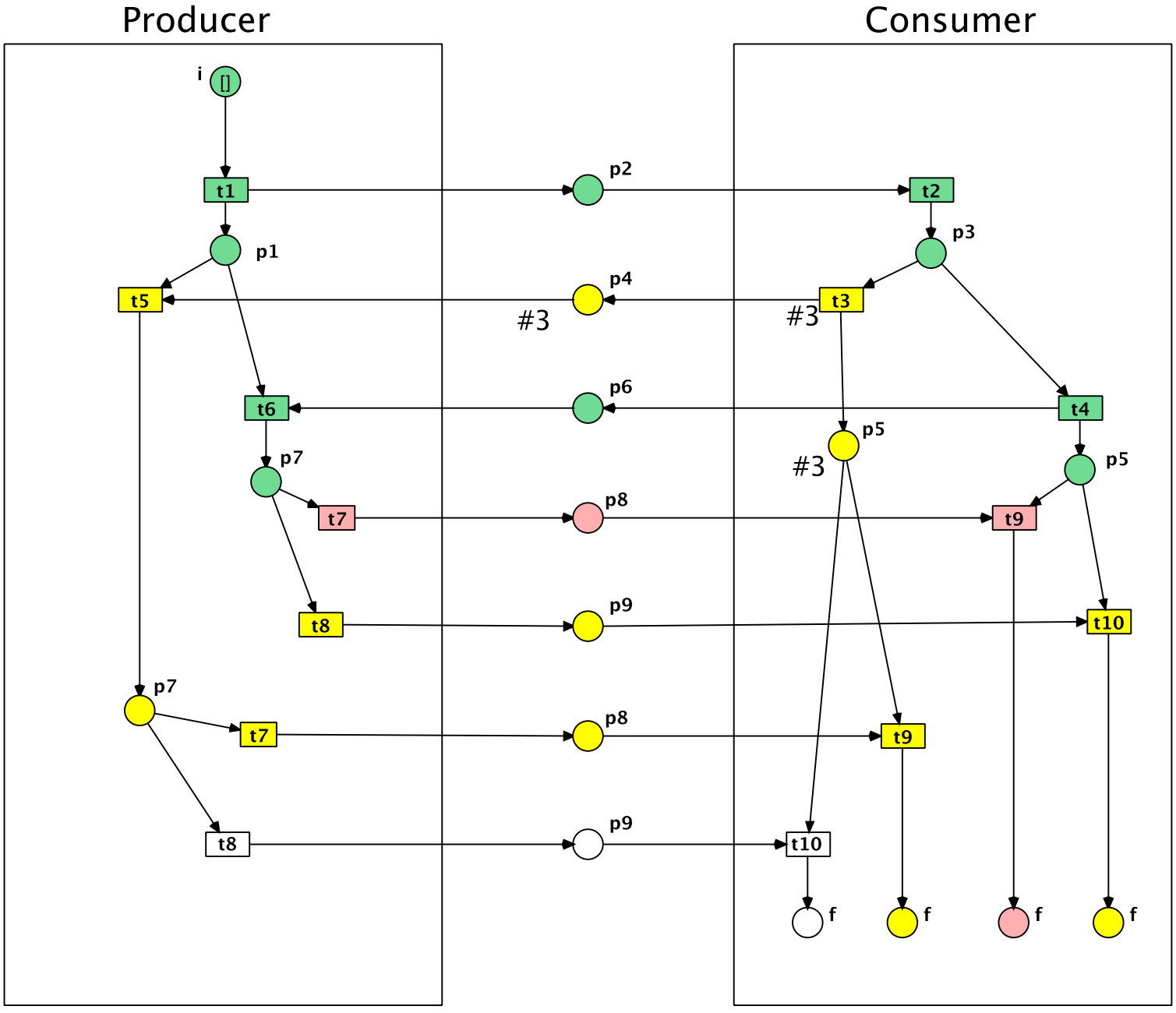
Negotiation in SONAR based on Unfoldings



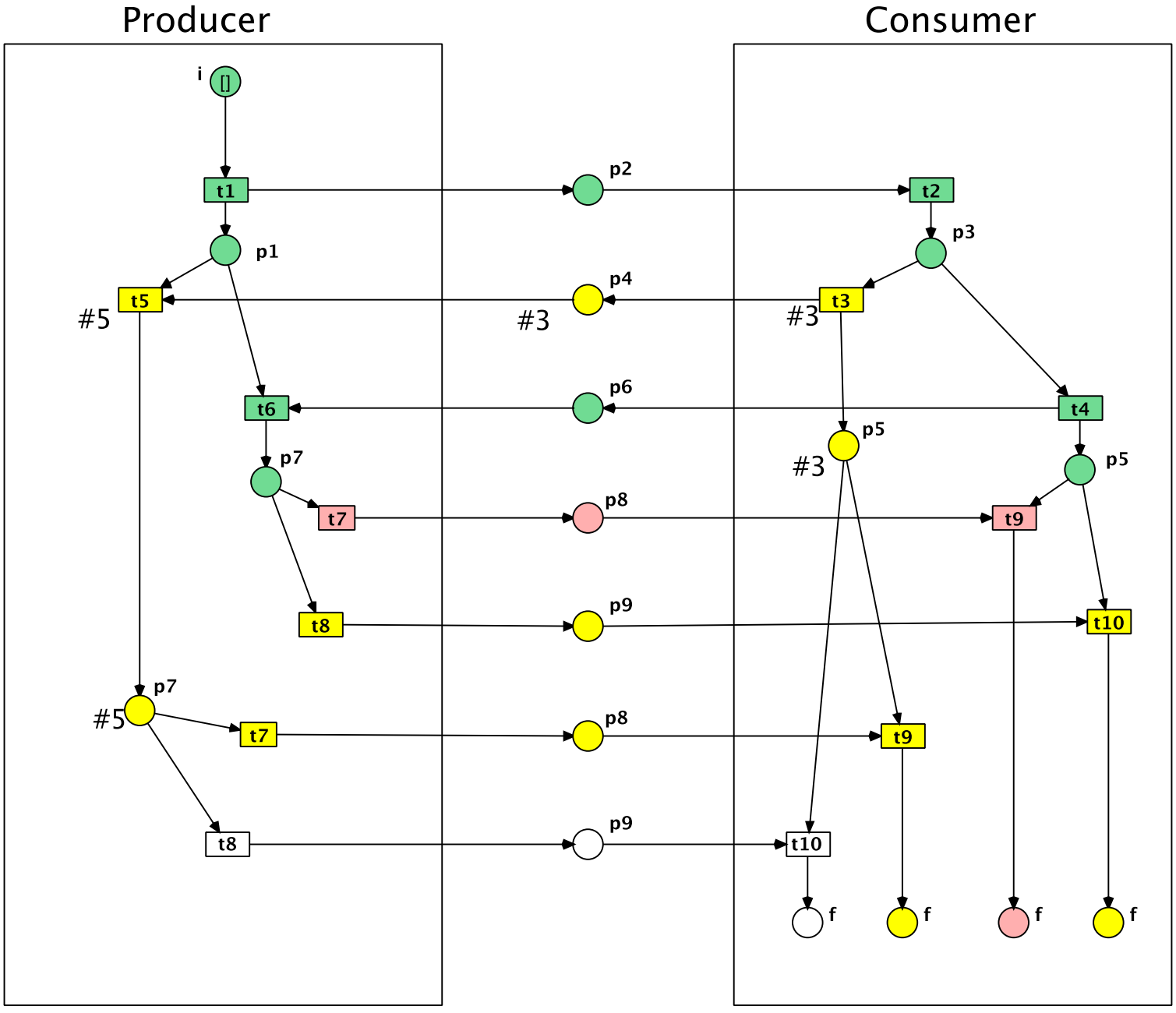
Negotiation in SONAR based on Unfoldings



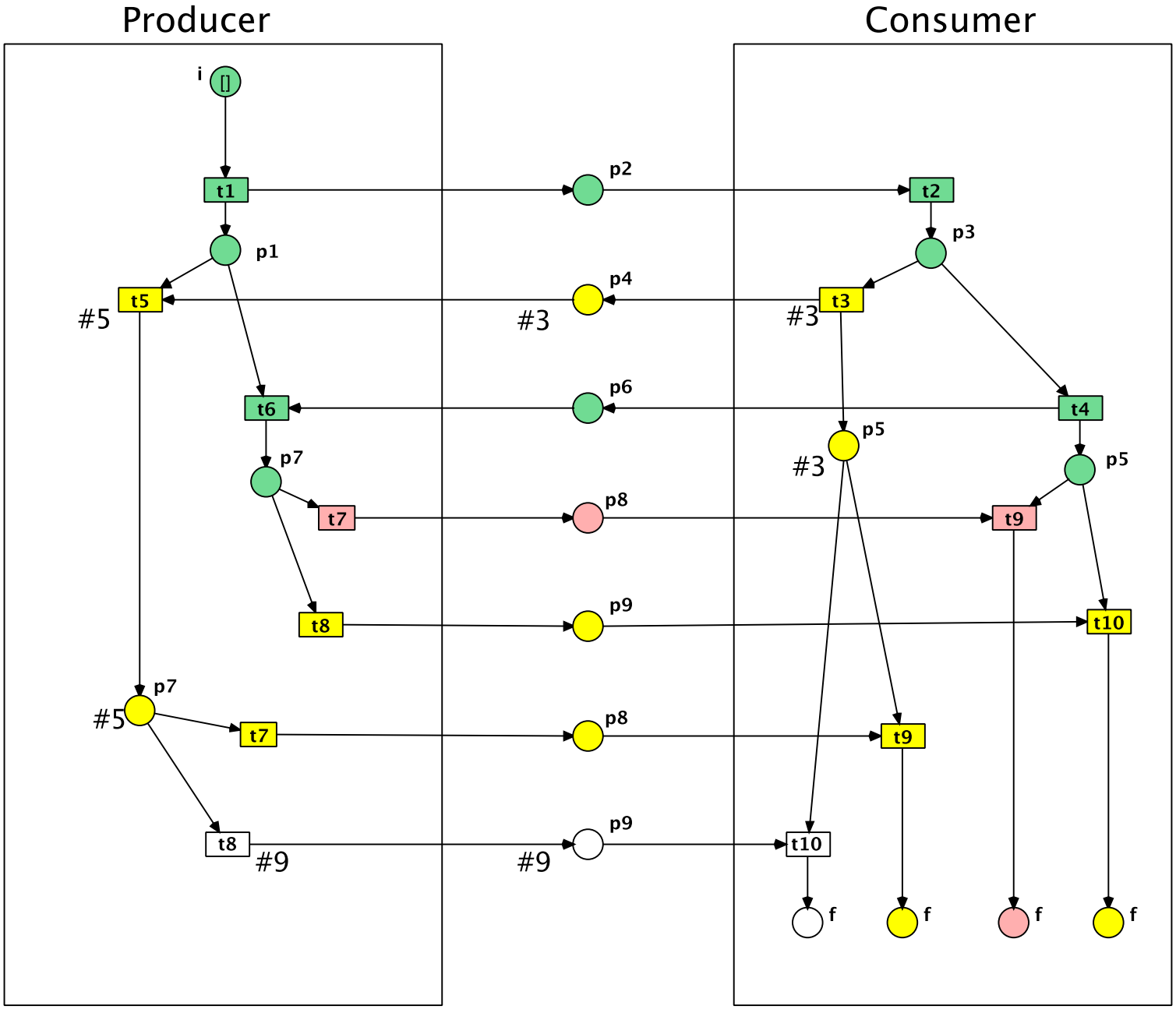
Negotiation in SONAR based on Unfoldings



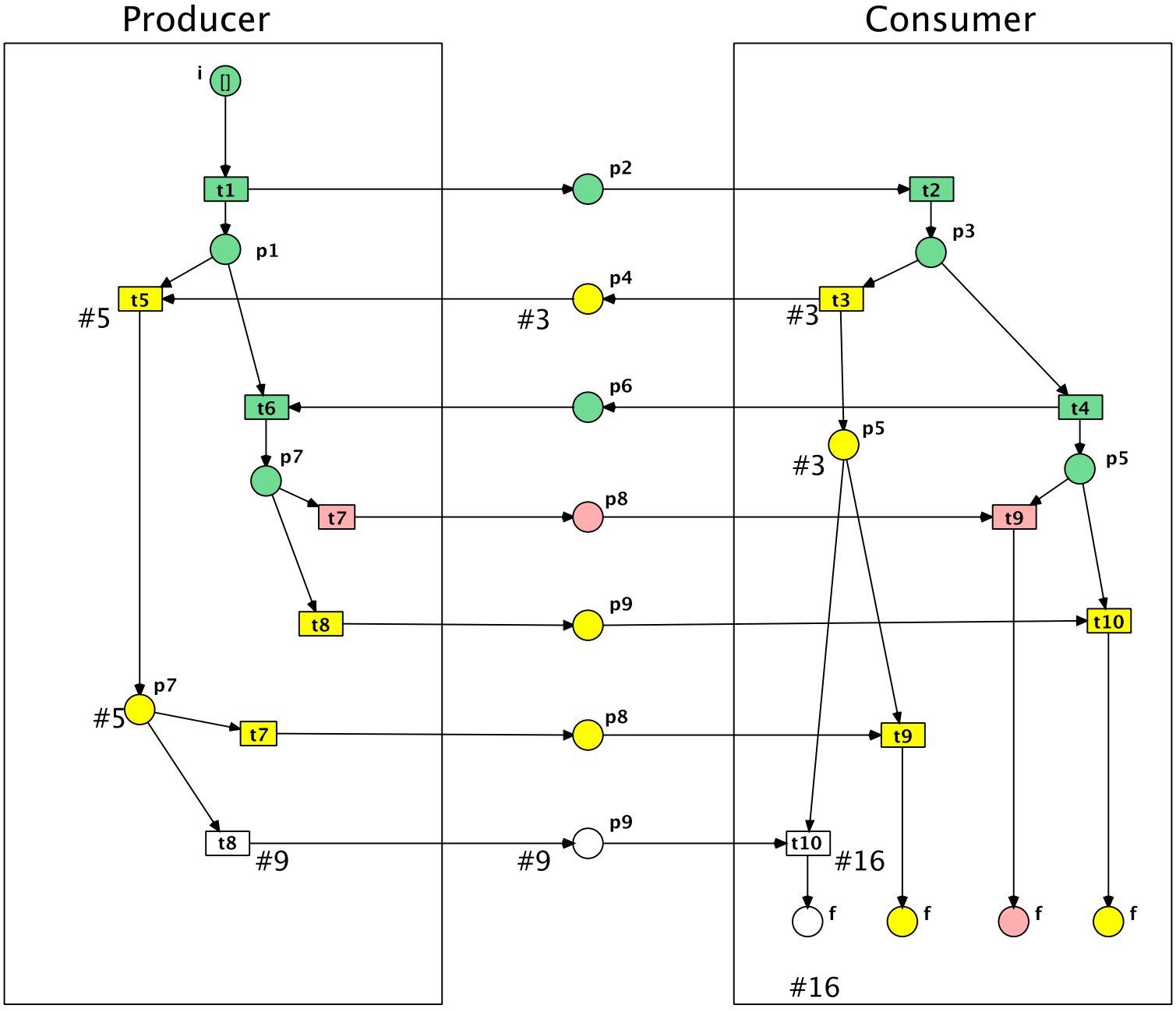
Negotiation in SONAR based on Unfoldings



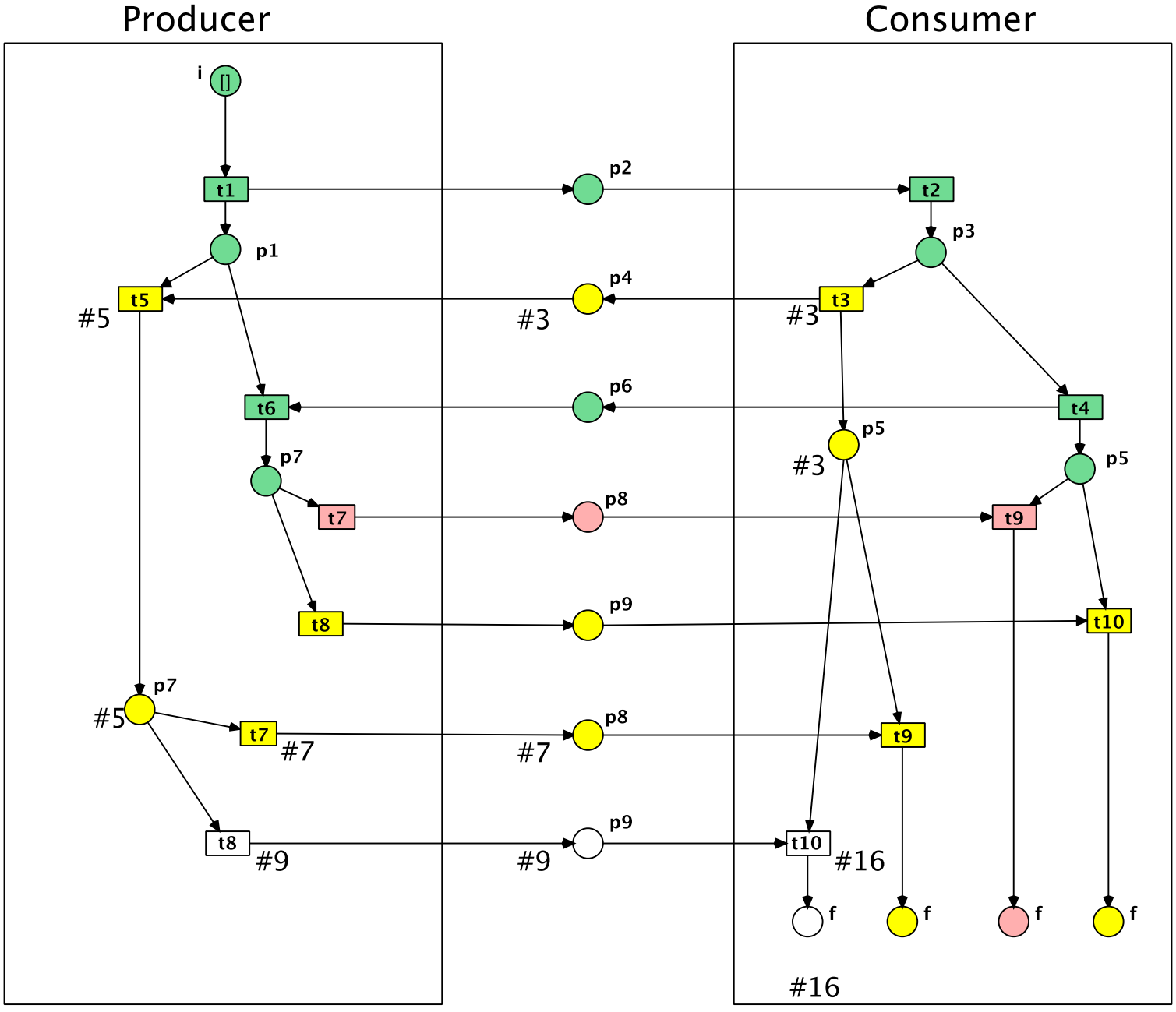
Negotiation in SONAR based on Unfoldings



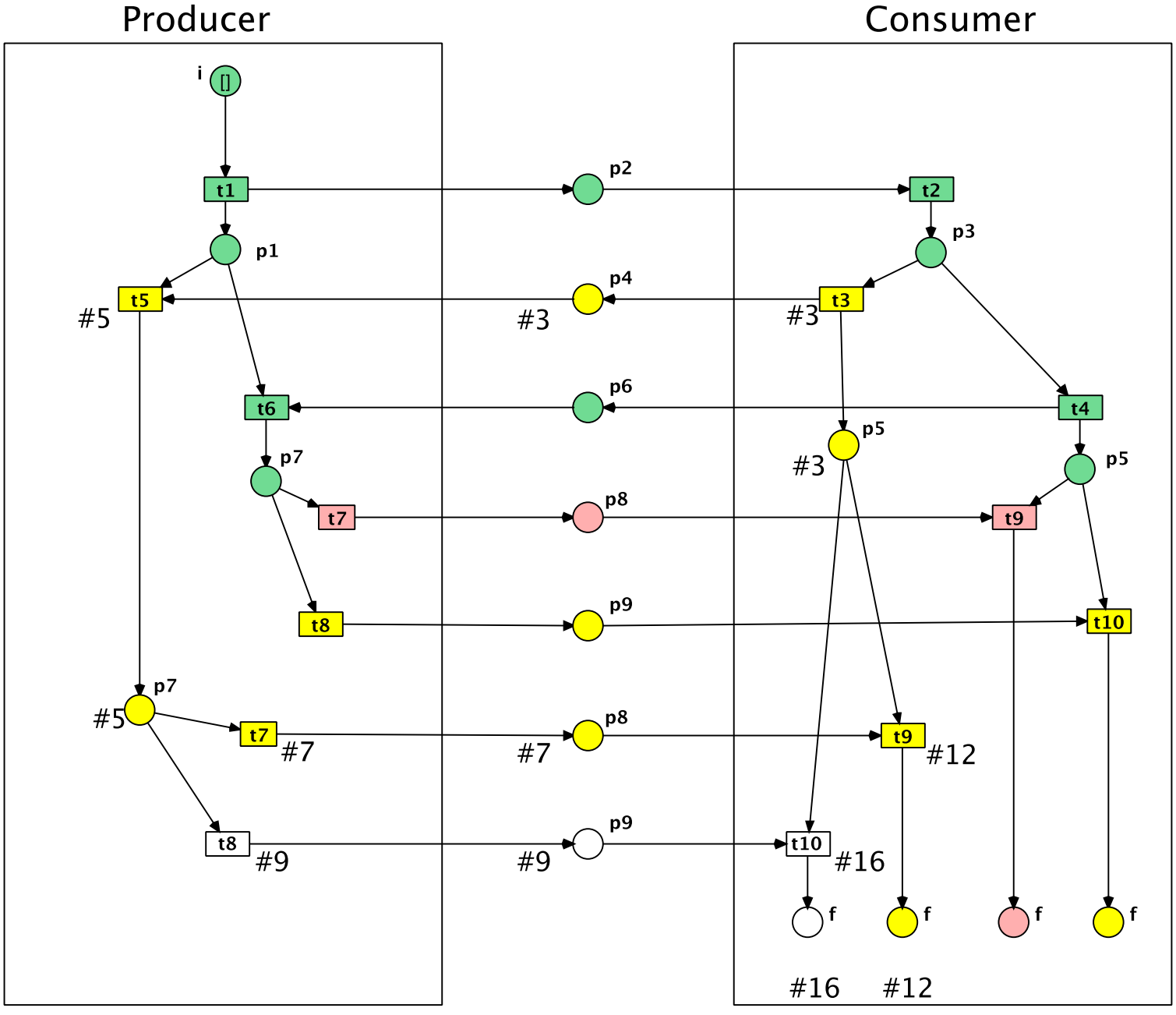
Negotiation in SONAR based on Unfoldings



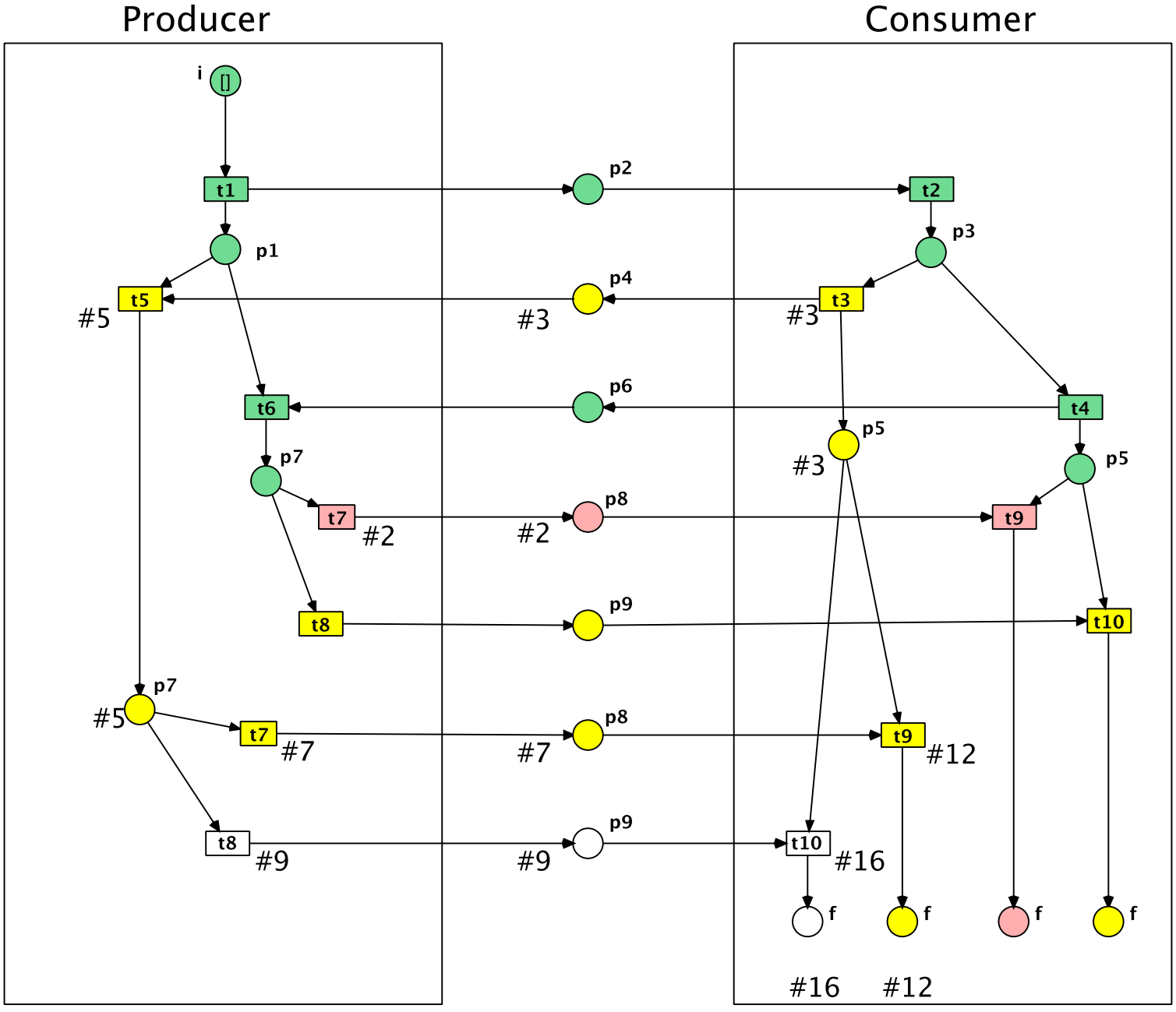
Negotiation in SONAR based on Unfoldings



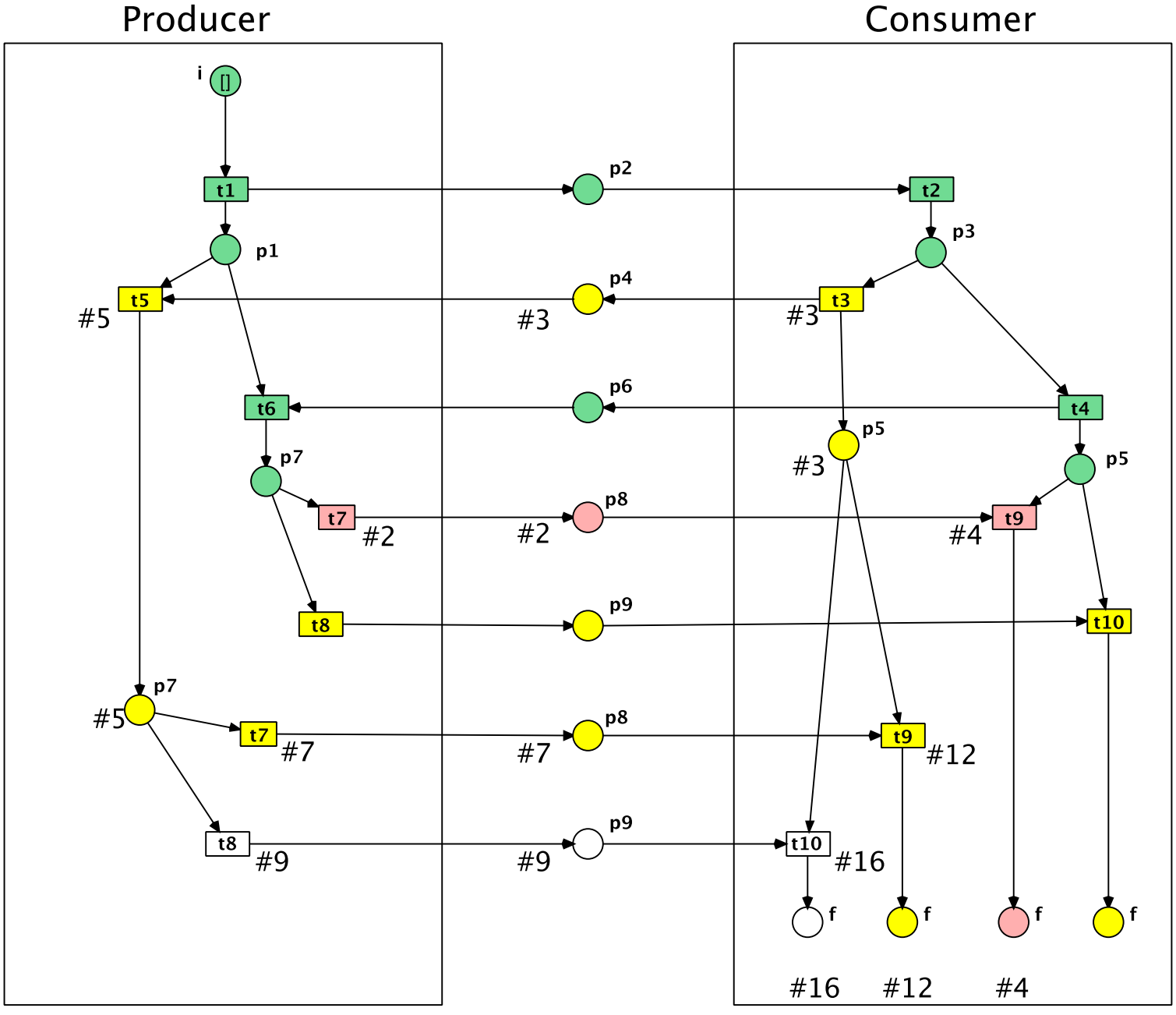
Negotiation in SONAR based on Unfoldings



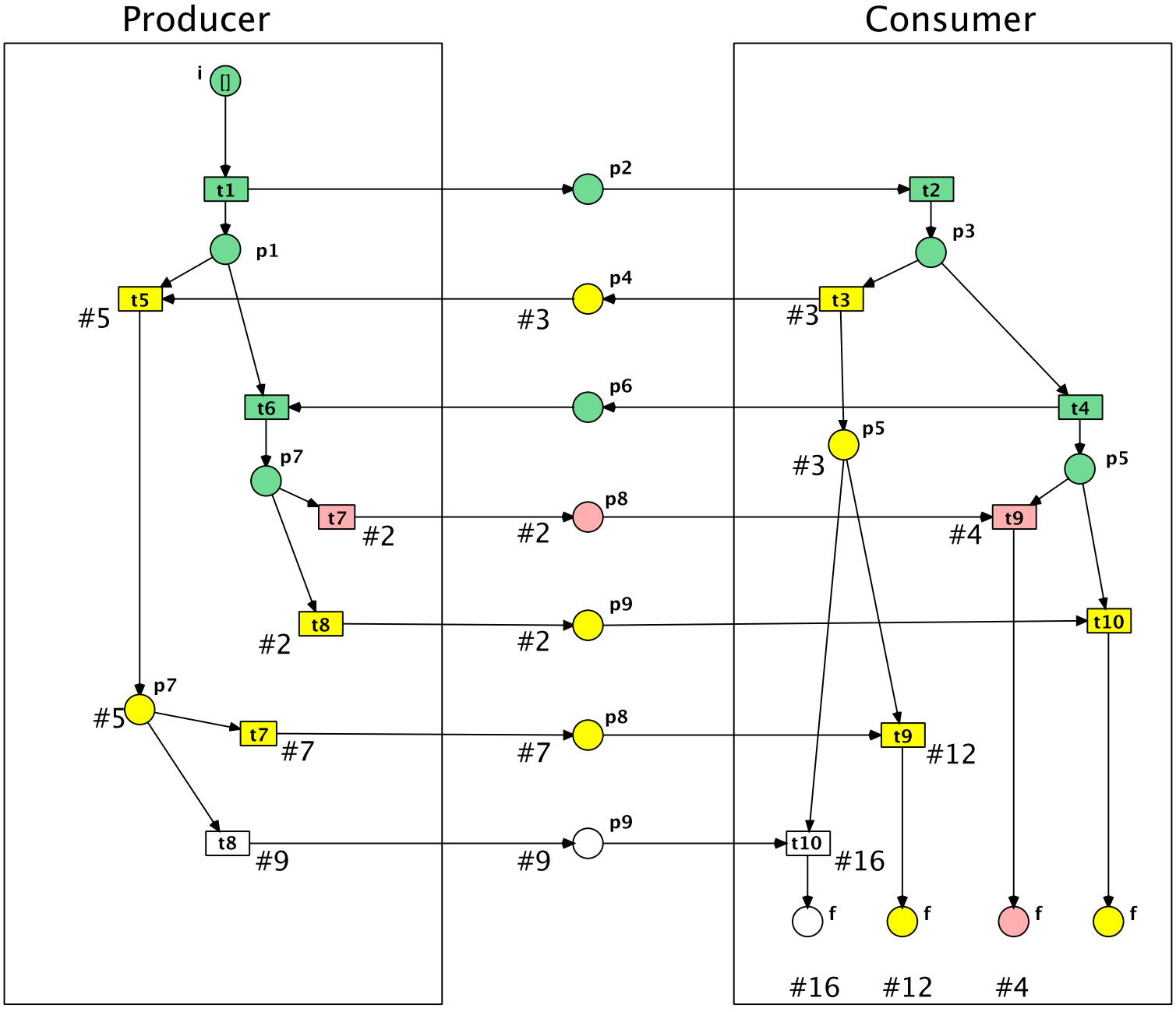
Negotiation in SONAR based on Unfoldings



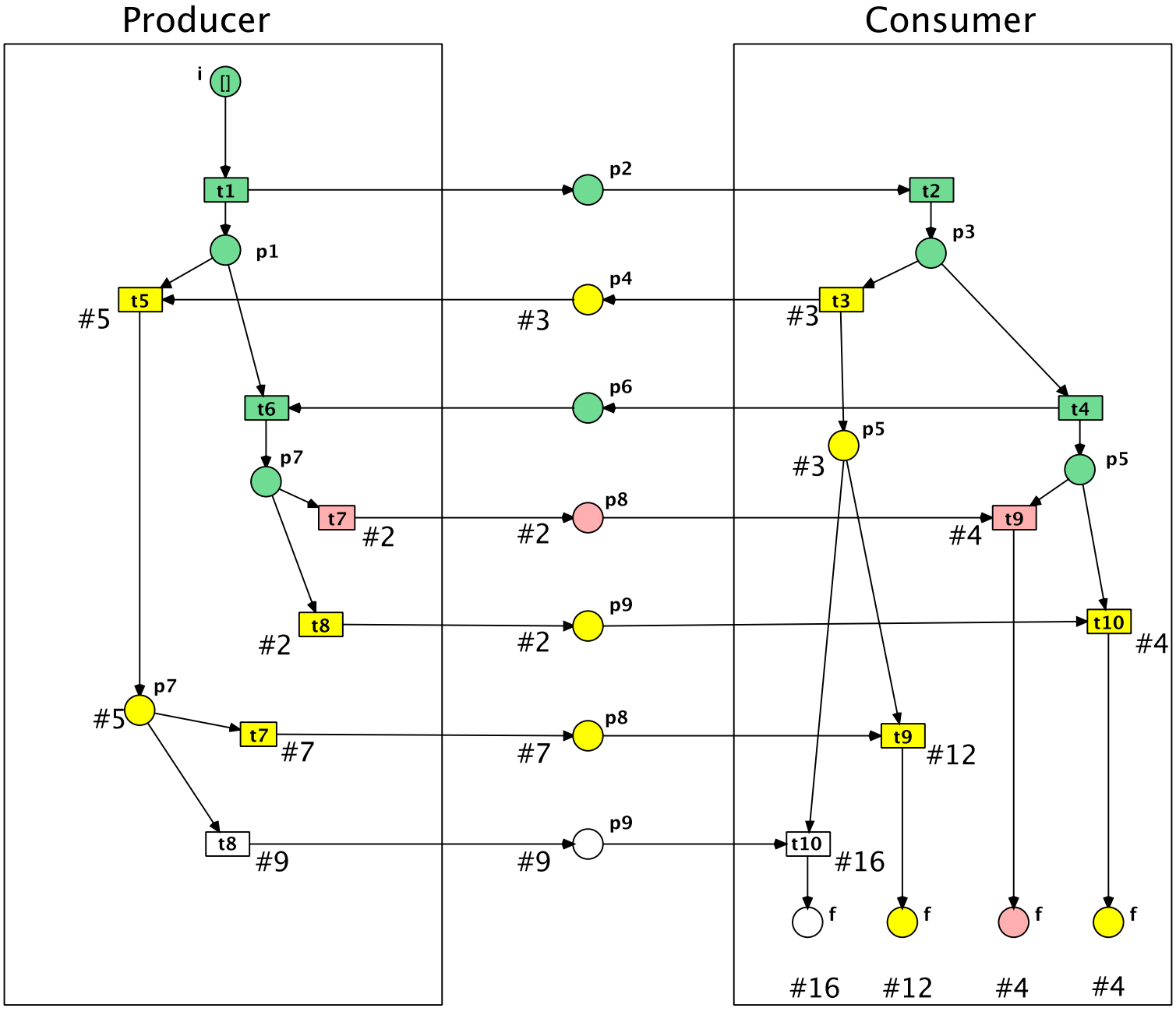
Negotiation in SONAR based on Unfoldings



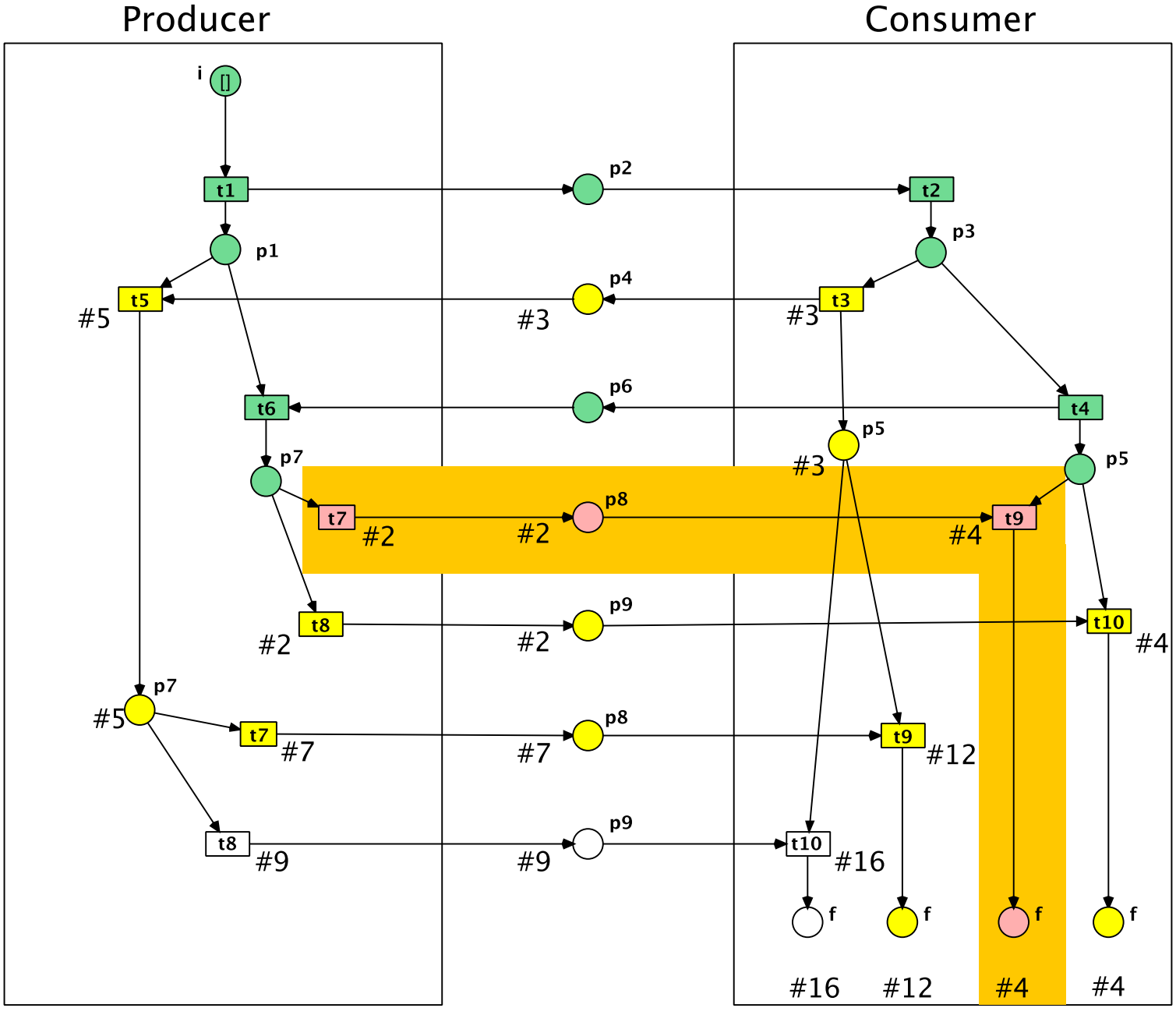
Negotiation in SONAR based on Unfoldings



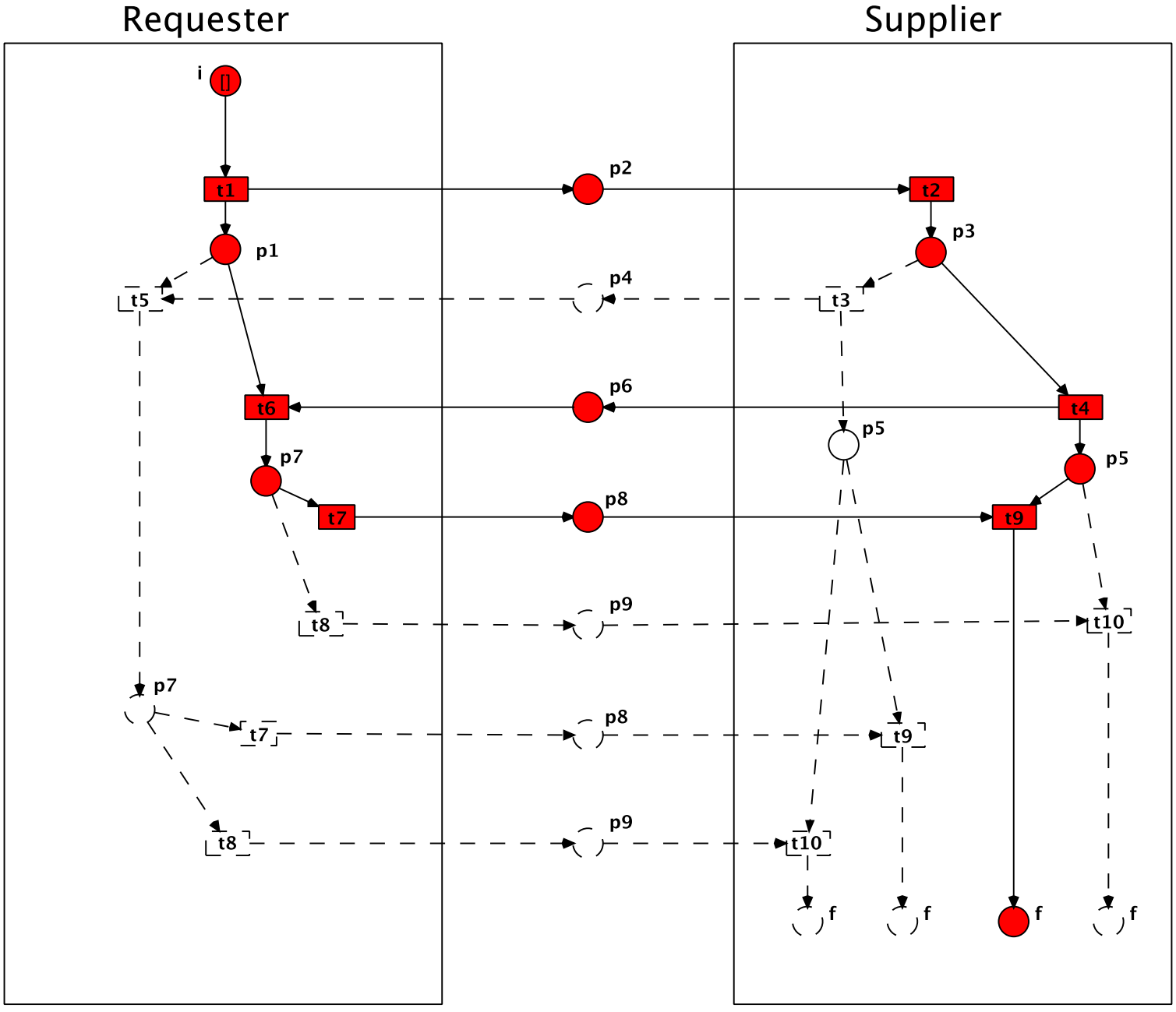
Negotiation in SONAR based on Unfoldings



Negotiation in SONAR based on Unfoldings

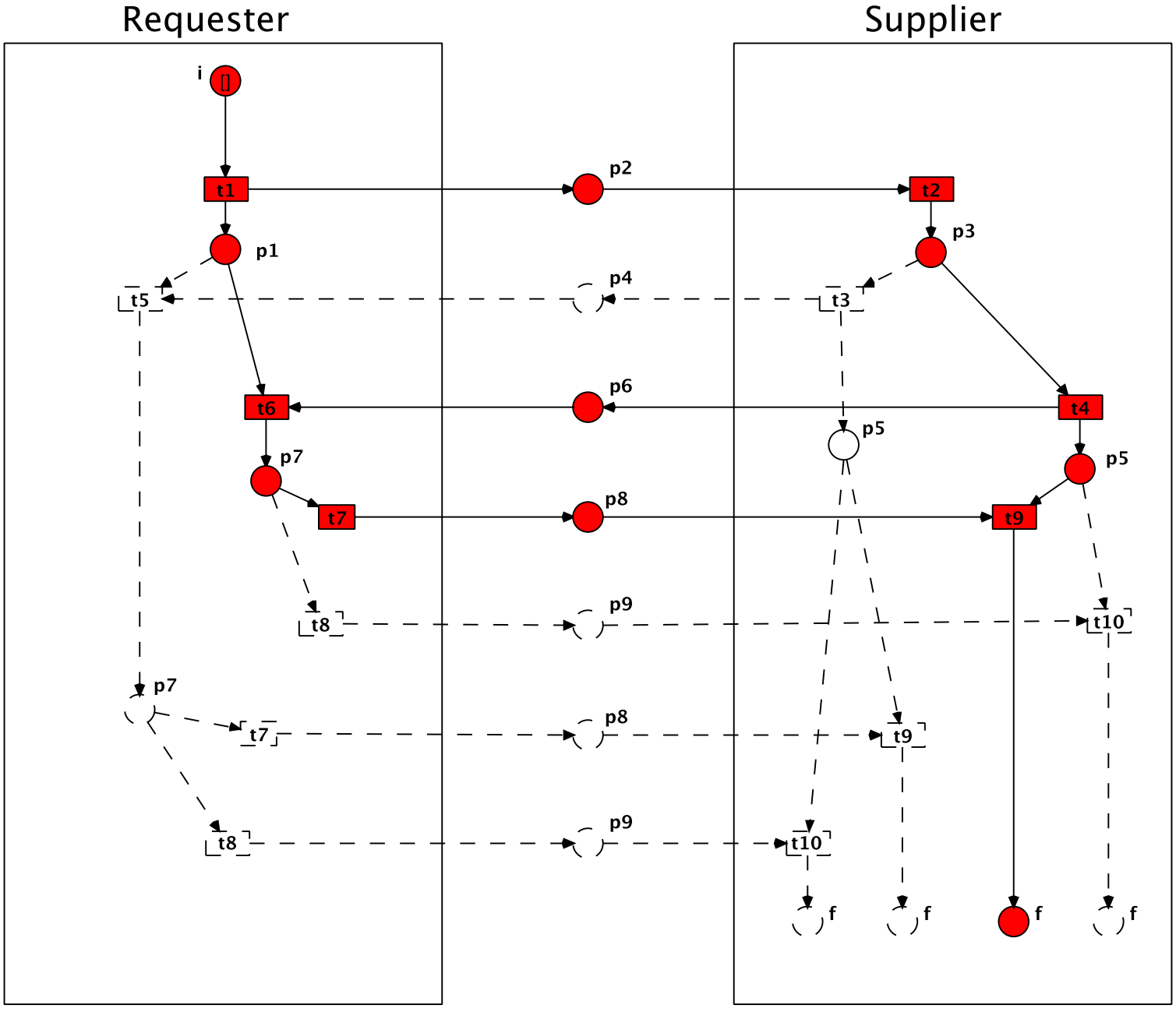


Negotiation in SONAR based on Unfoldings

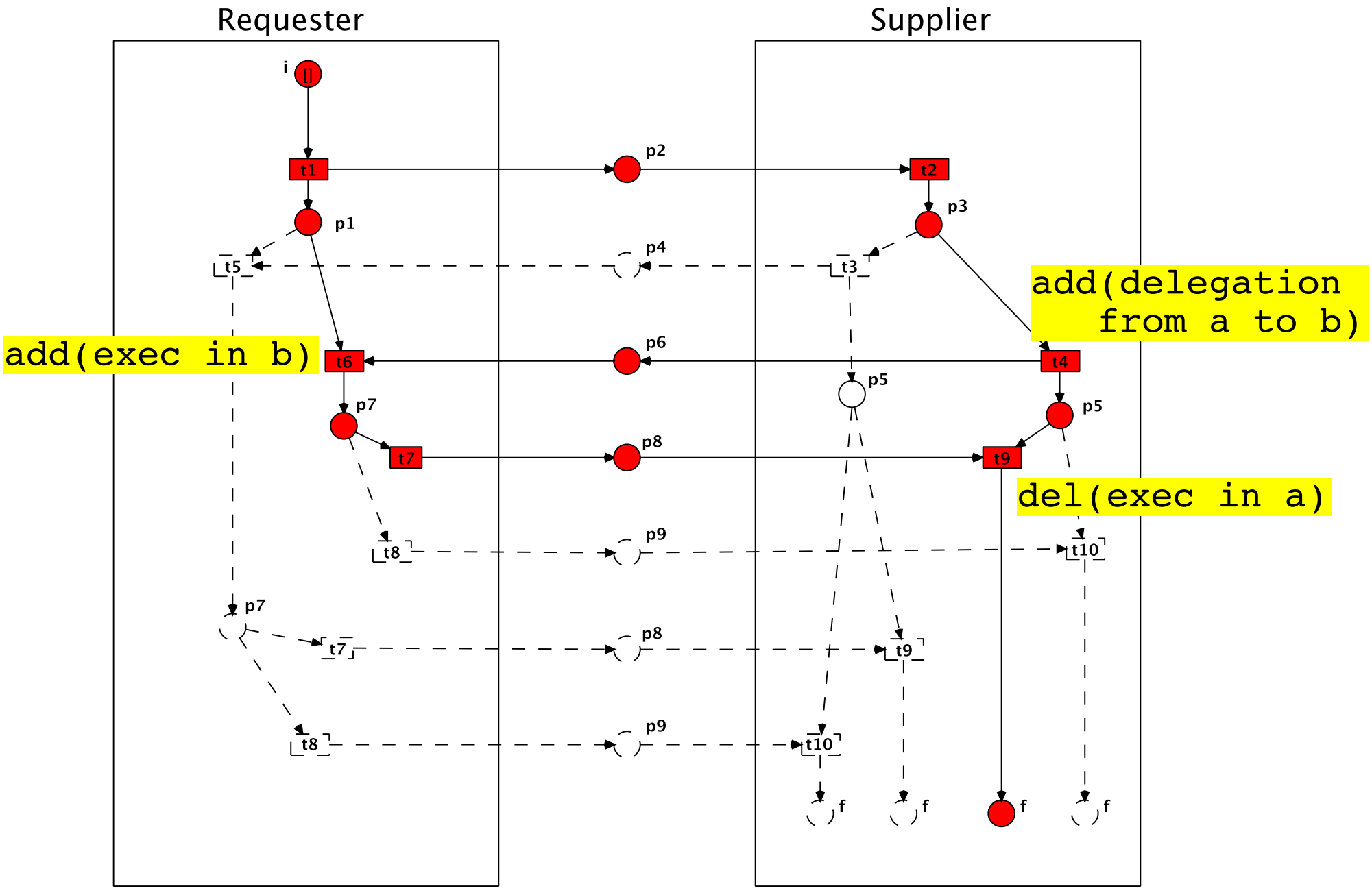


Outline

Execution of Plans in SONAR



Transformation τ induced by a SONAR Plan



Atomic Transformations of SONAR-Models

We allow two atomic transformations on organisations:

Definition

Define the operations $\mathtt{add}_t$ and $\mathtt{del}_t$ for each $t \in \mathcal{T}$ as:

$$\mathtt{add}_t(P,T,F) := \begin{cases} (P \cup t^\bullet, \quad T \cup \{t\}, \quad F \cup {}^\bullet t \times \{t\} \cup \{t\} \times t^\bullet) \\ \qquad\qquad \text{if } t \notin T \wedge {}^\bullet t \subseteq P \\ \text{undef.} \quad \text{otherwise} \end{cases}$$

$$\mathtt{del}_t(P,T,F) := \begin{cases} (P \setminus t^\bullet, \quad T \setminus \{t\}, \quad F \setminus ({}^\bullet t \times \{t\} \cup \{t\} \times t^\bullet)) \\ \qquad\qquad \text{if } t \in T \\ \text{undef.} \quad \text{otherwise} \end{cases}$$

Set of all atomic transformations: $ATF := \{\mathtt{add}_t, \mathtt{del}_t \mid t \in \mathcal{T}\}$.

Transformation: Reorganisation of the SONAR-Model

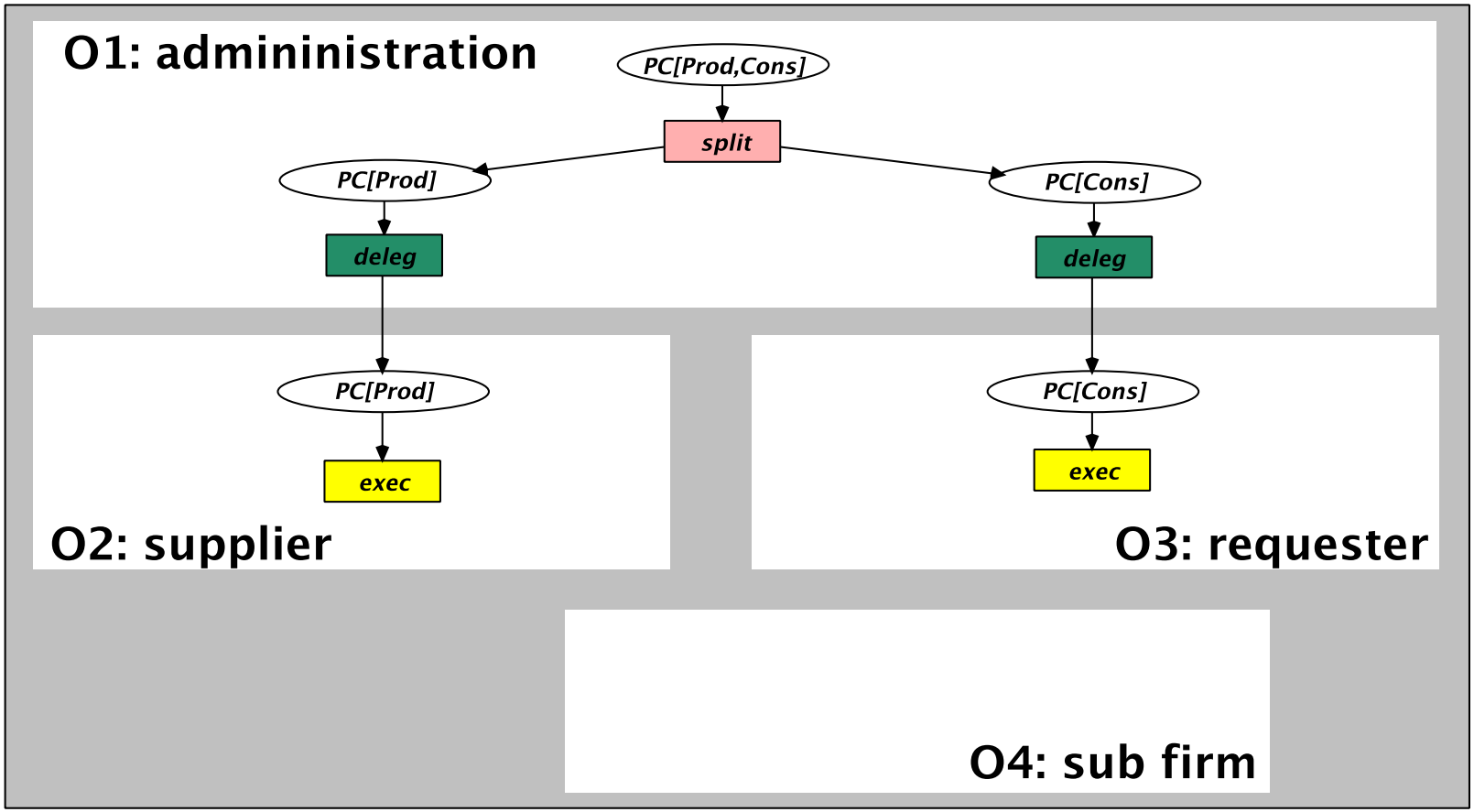
1. $\tau_1 =$ `add` *delegation* `from` O_3 `to` O_3
2. $\tau_2 =$ `add` *exec* `in` O_4
3. $\tau_3 =$ `del` *exec* `in` O_3

Question: Can we apply $\tau := \tau_1; \tau_2; \tau_3$ to the organisation?

Transformation: Reorganisation of the SONAR-Model

1. $\tau_1 = \texttt{add}\ \textit{delegation}\ \texttt{from}\ O_3\ \texttt{to}\ O_3$
2. $\tau_2 = \texttt{add}\ \textit{exec}\ \texttt{in}\ O_4$
3. $\tau_3 = \texttt{del}\ \textit{exec}\ \texttt{in}\ O_3$

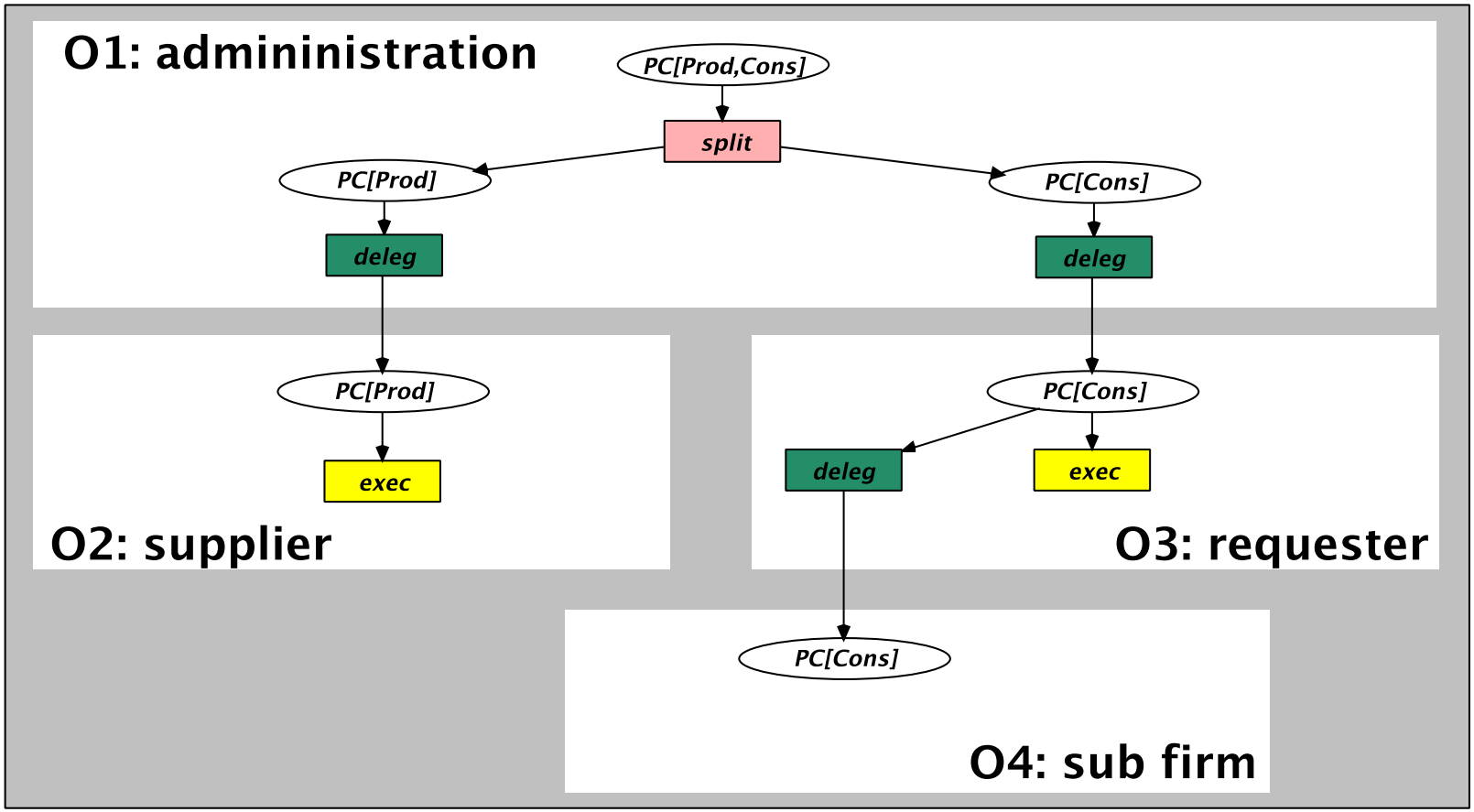
Question: Can we apply $\tau := \tau_1; \tau_2; \tau_3$ to the organisation?



Transformation: Reorganisation of the SONAR-Model

1. $\tau_1 = \texttt{add}\ \textit{delegation}\ \texttt{from}\ O_3\ \texttt{to}\ O_3$
2. $\tau_2 = \texttt{add}\ \textit{exec}\ \texttt{in}\ O_4$
3. $\tau_3 = \texttt{del}\ \textit{exec}\ \texttt{in}\ O_3$

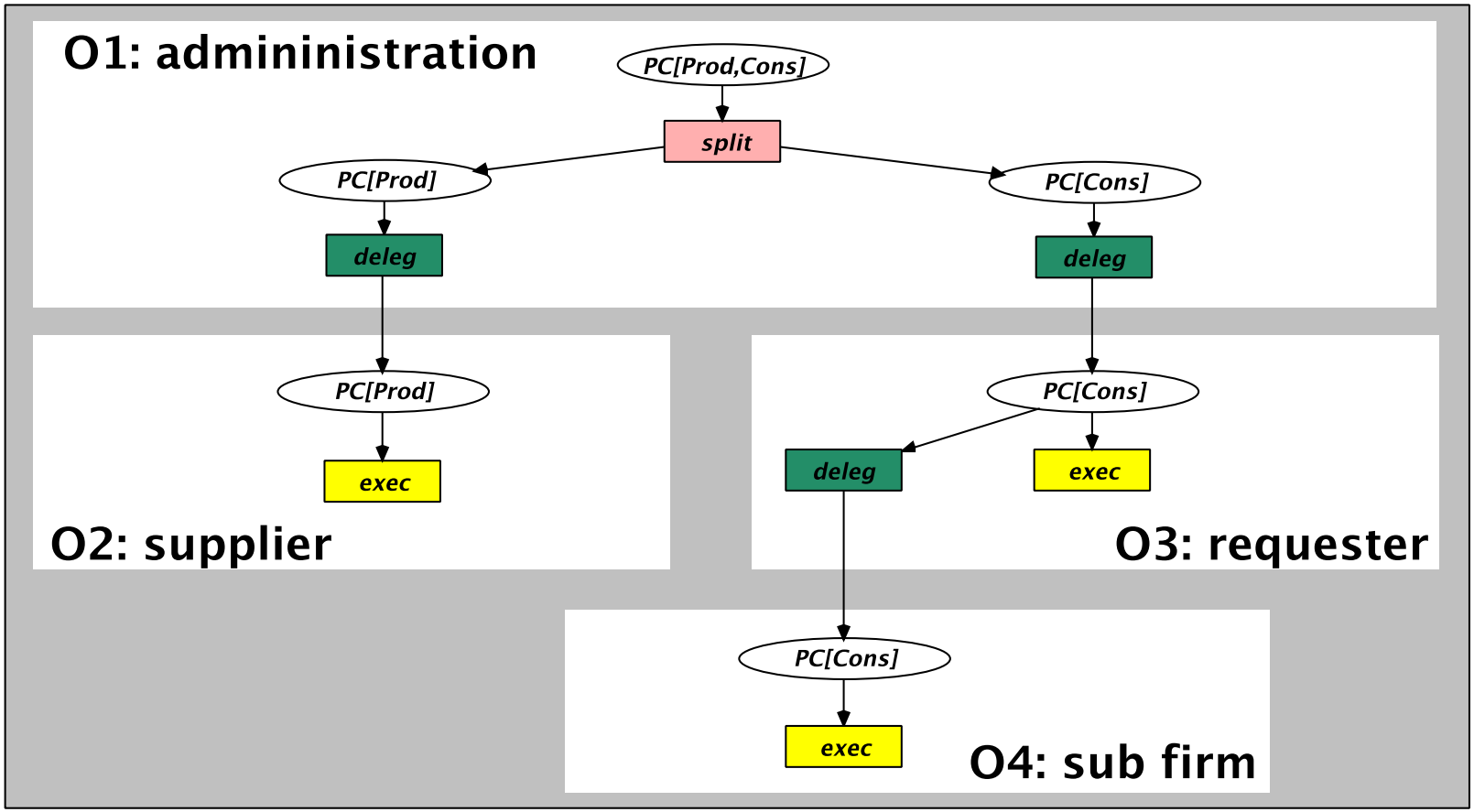
Question: Can we apply $\tau := \tau_1; \tau_2; \tau_3$ to the organisation?



Transformation: Reorganisation of the SONAR-Model

1. $\tau_1 = \texttt{add}\ \textit{delegation}\ \texttt{from}\ O_3\ \texttt{to}\ O_3$
2. $\tau_2 = \texttt{add}\ \textit{exec}\ \texttt{in}\ O_4$
3. $\tau_3 = \texttt{del}\ \textit{exec}\ \texttt{in}\ O_3$

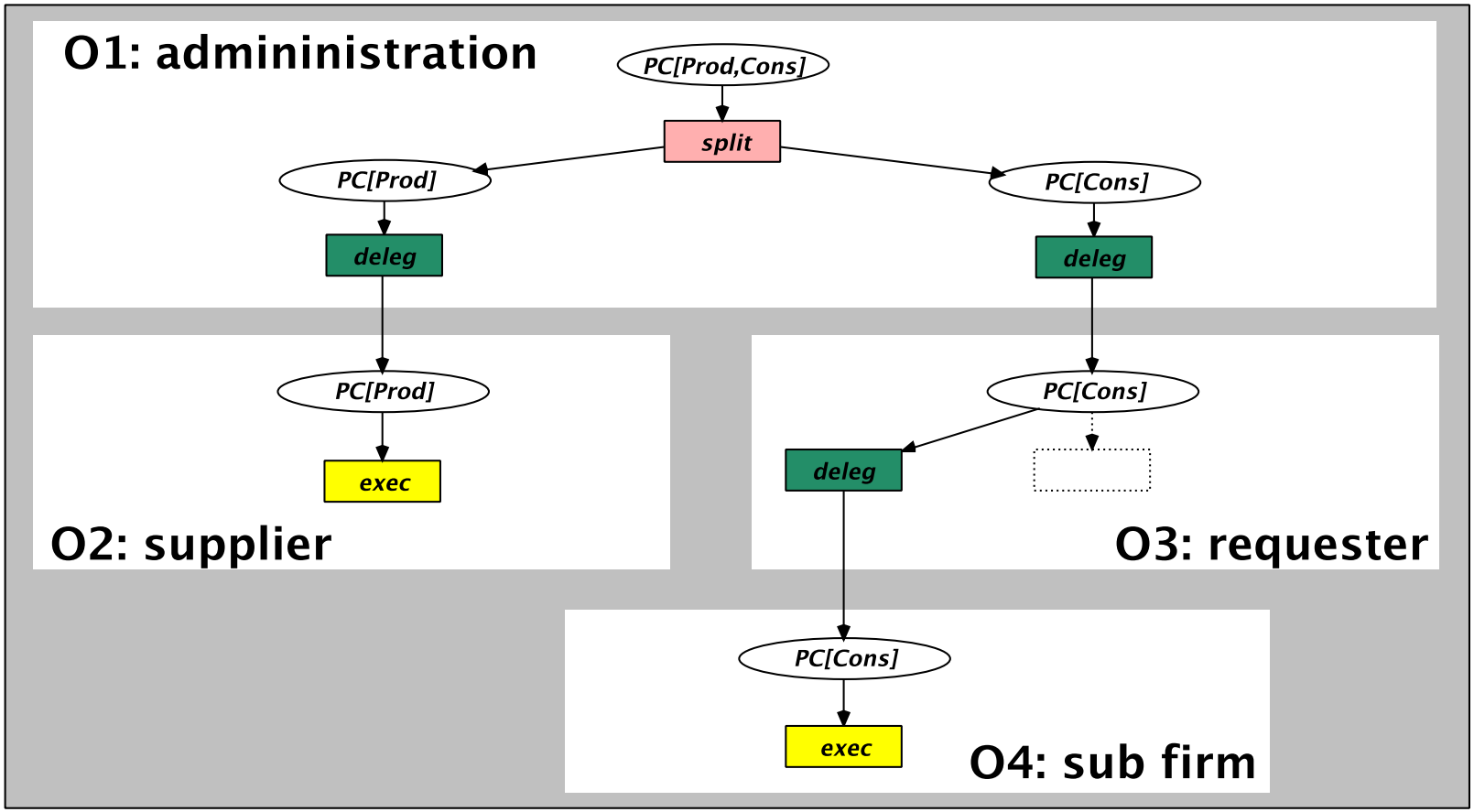
Question: Can we apply $\tau := \tau_1; \tau_2; \tau_3$ to the organisation?



Transformation: Reorganisation of the SONAR-Model

1. $\tau_1 = \texttt{add}\ \textit{delegation}\ \texttt{from}\ O_3\ \texttt{to}\ O_3$
2. $\tau_2 = \texttt{add}\ \textit{exec}\ \texttt{in}\ O_4$
3. $\tau_3 = \texttt{del}\ \textit{exec}\ \texttt{in}\ O_3$

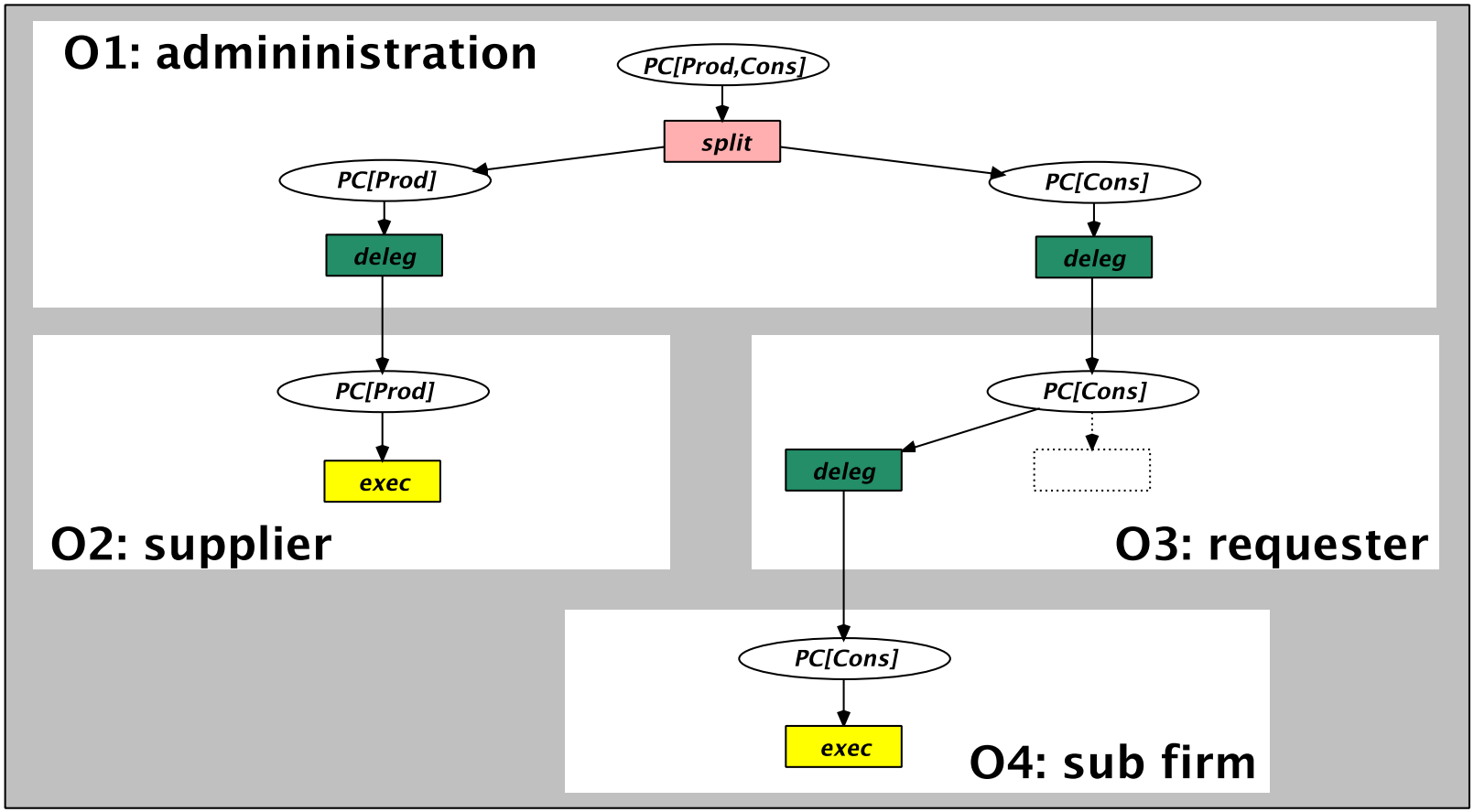
Question: Can we apply $\tau := \tau_1; \tau_2; \tau_3$ to the organisation?



Transformation: Reorganisation of the SONAR-Model

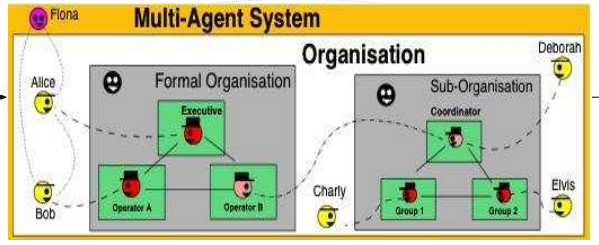
1. $\tau_1 = \texttt{add}\ \textit{delegation}\ \texttt{from}\ O_3\ \texttt{to}\ O_3$
2. $\tau_2 = \texttt{add}\ \textit{exec}\ \texttt{in}\ O_4$
3. $\tau_3 = \texttt{del}\ \textit{exec}\ \texttt{in}\ O_3$

Question: Can we apply $\tau := \tau_1; \tau_2; \tau_3$ to the organisation? Yes, we can!



Summary: Overview of SONAR Concepts

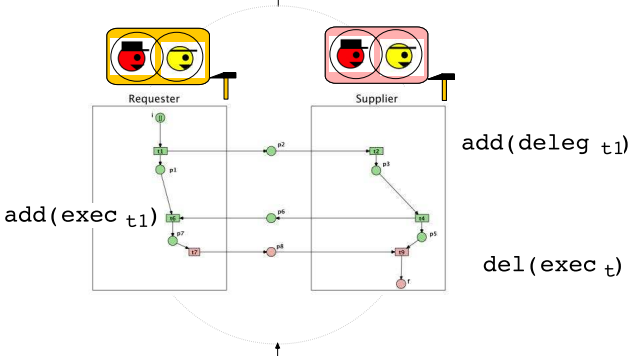
reorganise



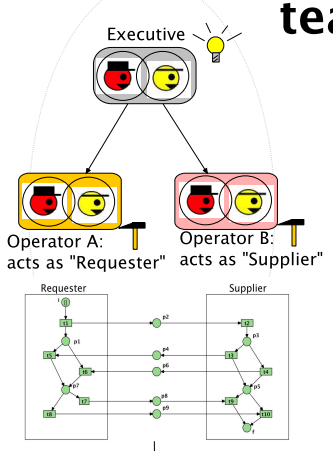
form team

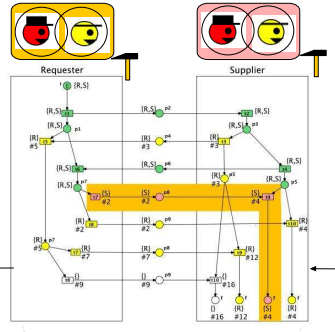
Organsiation-MAS

transformation



team





team plan

execute

negotiate

Outline

Problems

Problems:

- Can one ensure that the transformation (generated by a team plan) is applicable to the current organisation?
- Can one ensure that the resulting organisation is well formed?
- Can one guide the negotiation process that applicability is guaranteed?

Our approach in SONAR:

- Study the space of transformations.
- Here: Specify transformations as processes of a special Petri net, called the **meta organisation** $\hat{N}$.
  - A meta-marking $\hat{m}$ encodes an organisation net $N := N(\hat{m})$.
  - Each meta-transition $\hat{t}$ either activates or deactivates an implementation $t \in \mathcal{T}$.
- Approach is similar to *graph grammar processes*, but simpler.

Problems

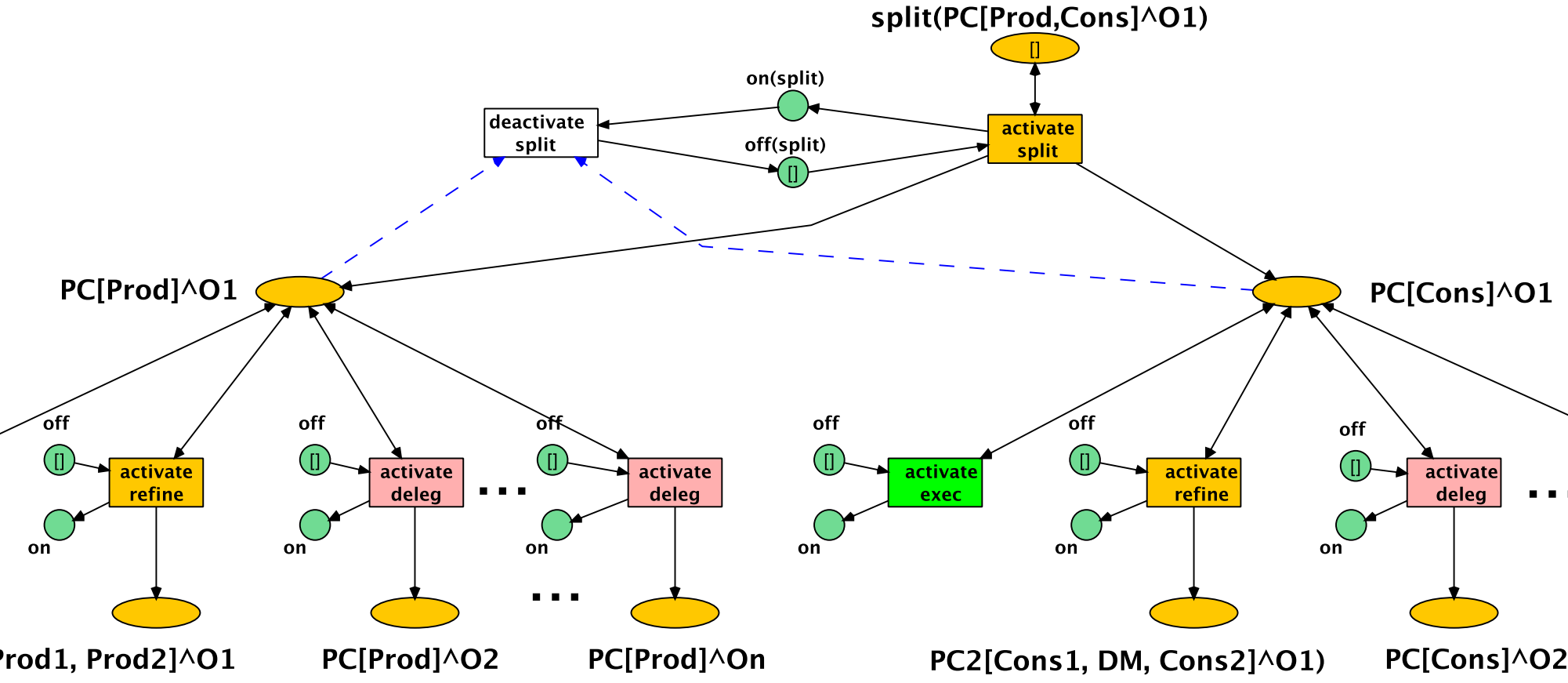
Problems:

- Can one ensure that the transformation (generated by a team plan) is applicable to the current organisation?
- Can one ensure that the resulting organisation is well formed?
- Can one guide the negotiation process that applicability is guaranteed?

Our approach in SONAR:

- Study the space of transformations.
- Here: Specify transformations as processes of a special Petri net, called the **meta organisation** $\hat{N}$.
  - A meta-marking $\hat{m}$ encodes an organisation net $N := N(\hat{m})$.
  - Each meta-transition $\hat{t}$ either activates or deactivates an implementation $t \in \mathcal{T}$.
- Approach is similar to *graph grammar processes*, but simpler.

Meta Organisation: Example



Simulation

Define the isomorphism $h : (\hat{T} \cup \{\perp\})^* \to ATF^*$ as:

$h(\texttt{activate}_t) = \texttt{add}_t, \quad h(\texttt{deactivate}_t) = \texttt{del}_t, \quad \text{and} \quad h(\perp) = id.$

Theorem

Meta-Events are activated transformations:

$$\begin{array}{ccc} \hat{m} & \xrightarrow{\hat{t}} & \hat{m}' \\ \downarrow{\scriptstyle N(\cdot)} & & \downarrow{\scriptstyle N(\cdot)} \\ N = N(\hat{m}) & \xrightarrow{h(\hat{t})} & N' = N(\hat{m}') \end{array}$$

Application to Distributed Team Planning

- How can one guarantee that the transformation generated by the team plan is applicable to the organisation?
- Answer: We *synchronise* the team-protocol D with the meta-organisation $\hat{N}_{P^0}$.
- Define the *synchronous product* $D \otimes \hat{N}_{P^0}$.
  - A protocol transition t_D labelled with $\texttt{add}_t$ is synchronised with the meta-transition $\texttt{activate}_t$.
  - Them, t_D is activated only if the transformation $\texttt{add}_t$ is applicable to the current organisation.
- Negotiation constructs a partial plan for the product $(D \otimes \hat{N}_{P^0})$ – and not only for the team-protocol D.
- Then, each team plan for $(D \otimes \hat{N}_{P^0})$ generates an applicable transformation *by construction*.

Outline

Analysis of the Meta-Organisation

A **policy** Φ is a predicate on organisation nets N.

Validity of a policy Φ for the organisation N is denoted $N \models \Phi$.

Define the set of meta-markings that satisfy a policy Φ as:

$$SAT(\Phi) := \{\hat{m} \in RS(\hat{N}_{P^0}, \hat{m}_0) \mid N(\hat{m}) \models \Phi\}$$

Definition

The meta-organisation $\hat{N}$ *enforces* the policy Φ if $SAT(\Phi) = RS(\hat{N}_{P^0}, \hat{m}_0)$.

Assume, that the set of agents $\mathcal{A}$ and protocols $\mathcal{D}$ are finite sets.

Theorem

Given a meta-organisation $\hat{N}$, it can be checked (using standard model checking techniques), whether the policy is enforced.

Note, that whenever $\mathcal{A}$ and $\mathcal{D}$ are finite sets, then $\hat{N}$ and its state space are finite, too.

Analysis of the Meta-Organisation: Stability

But, in almost all cases a meta-organisation $\hat{N}$ does *not* enforce a policy Φ.

Usually, this is not problematic, since it is not necessary that each $N(\hat{m})$ satisfy Φ after each *step* of the team plan;

Instead: It is necessary that for each reachable meta-marking $\hat{m}$ we have the possibility to satisfy Φ again:

$$\forall \hat{m} \in RS(\hat{N}_{P^0}, \hat{m}_0) : \exists \hat{m}' \in RS(\hat{N}_{P^0}, \hat{m}) : \hat{m}' \in SAT(\Phi)$$

But, the answer is trivially “yes”, since each transformation can be undone (since $\hat{N}$ is revertible).

Therefore we exclude the $\texttt{deactivate}_t$ transitions from the analysis:

Definition

The meta-organisation $\hat{N}$ is **stable** w.r.t. the policy Φ if we have for $\hat{A} = \{\texttt{deactivate}_t \mid t \in \mathcal{T}\}$:

$$\forall \hat{m} \in RS(\hat{N}_{P^0} - \hat{A}, \hat{m}_0) : \exists \hat{m}' \in RS(\hat{N}_{P^0} - \hat{A}, \hat{m}) : \hat{m}' \in SAT(\Phi)$$

Analysis of the Meta-Organisation: Stability

But, in almost all cases a meta-organisation $\hat{N}$ does *not* enforce a policy Φ.

Usually, this is not problematic, since it is not necessary that each $N(\hat{m})$ satisfy Φ after each *step* of the team plan;

Instead: It is necessary that for each reachable meta-marking $\hat{m}$ we have the possibility to satisfy Φ again:

$$\forall \hat{m} \in RS(\hat{N}_{P^0}, \hat{m}_0) : \exists \hat{m}' \in RS(\hat{N}_{P^0}, \hat{m}) : \hat{m}' \in SAT(\Phi)$$

But, the answer is trivially “yes”, since each transformation can be undone (since $\hat{N}$ is revertible).

Therefore we exclude the $\texttt{deactivate}_t$ transitions from the analysis:

Definition

The meta-organisation $\hat{N}$ is **stable** w.r.t. the policy Φ if we have for $\hat{A} = \{\texttt{deactivate}_t \mid t \in \mathcal{T}\}$:

$$\forall \hat{m} \in RS(\hat{N}_{P^0} - \hat{A}, \hat{m}_0) : \exists \hat{m}' \in RS(\hat{N}_{P^0} - \hat{A}, \hat{m}) : \hat{m}' \in SAT(\Phi)$$

Analysis of the Meta-Organisation: Stability

But, in almost all cases a meta-organisation $\hat{N}$ does *not* enforce a policy Φ.

Usually, this is not problematic, since it is not necessary that each $N(\hat{m})$ satisfy Φ after each *step* of the team plan;

Instead: It is necessary that for each reachable meta-marking $\hat{m}$ we have the possibility to satisfy Φ again:

$$\forall \hat{m} \in RS(\hat{N}_{P^0}, \hat{m}_0) : \exists \hat{m}' \in RS(\hat{N}_{P^0}, \hat{m}) : \hat{m}' \in SAT(\Phi)$$

But, the answer is trivially “yes”, since each transformation can be undone (since $\hat{N}$ is revertible).

Therefore we exclude the $\texttt{deactivate}_t$ transitions from the analysis:

Definition

The meta-organisation $\hat{N}$ is **stable** w.r.t. the policy Φ if we have for $\hat{A} = \{\texttt{deactivate}_t \mid t \in \mathcal{T}\}$:

$$\forall \hat{m} \in RS(\hat{N}_{P^0} - \hat{A}, \hat{m}_0) : \exists \hat{m}' \in RS(\hat{N}_{P^0} - \hat{A}, \hat{m}) : \hat{m}' \in SAT(\Phi)$$

Analysis of the Meta-Organisation: Stability

But, in almost all cases a meta-organisation $\hat{N}$ does *not* enforce a policy Φ.

Usually, this is not problematic, since it is not necessary that each $N(\hat{m})$ satisfy Φ after each *step* of the team plan;

Instead: It is necessary that for each reachable meta-marking $\hat{m}$ we have the possibility to satisfy Φ again:

$$\forall \hat{m} \in RS(\hat{N}_{P^0}, \hat{m}_0) : \exists \hat{m}' \in RS(\hat{N}_{P^0}, \hat{m}) : \hat{m}' \in SAT(\Phi)$$

But, the answer is trivially “yes”, since each transformation can be undone (since $\hat{N}$ is revertible).

Therefore we exclude the $\texttt{deactivate}_t$ transitions from the analysis:

Definition

The meta-organisation $\hat{N}$ is **stable** w.r.t. the policy Φ if we have for $\hat{A} = \{\texttt{deactivate}_t \mid t \in \mathcal{T}\}$:

$$\forall \hat{m} \in RS(\hat{N}_{P^0} - \hat{A}, \hat{m}_0) : \exists \hat{m}' \in RS(\hat{N}_{P^0} - \hat{A}, \hat{m}) : \hat{m}' \in SAT(\Phi)$$

Analysis of the Meta-Organisation: Stability

Definition

The meta-organisation $\hat{N}$ is **stable** w.r.t. the policy Φ if we have for $\hat{A} = \{\texttt{deactivate}_t \mid t \in \mathcal{T}\}$:

$$\forall \hat{m} \in RS(\hat{N}_{P^0} - \hat{A}, \hat{m}_0) : \exists \hat{m}' \in RS(\hat{N}_{P^0} - \hat{A}, \hat{m}) : \hat{m}' \in SAT(\Phi)$$

Theorem

Given a meta-organisation $\hat{N}$, it can be checked using standard model checking techniques, whether a policy Φ is stable.

Proof.

Stability is a kind of *liveness property* and can be checked by computing the SCCs of the state space:
Check whether each terminal SCC contains an $\hat{m}$ that satisfies Φ. □

Outline