# Process Oriented Data Science

UNIVERSITAT POLITÈCNICA
DE CATALUNYA
BARCELONATECH

*Campus d'Excel·lència Internacional*

Josep Carmona
Computer Science Department

# Outline

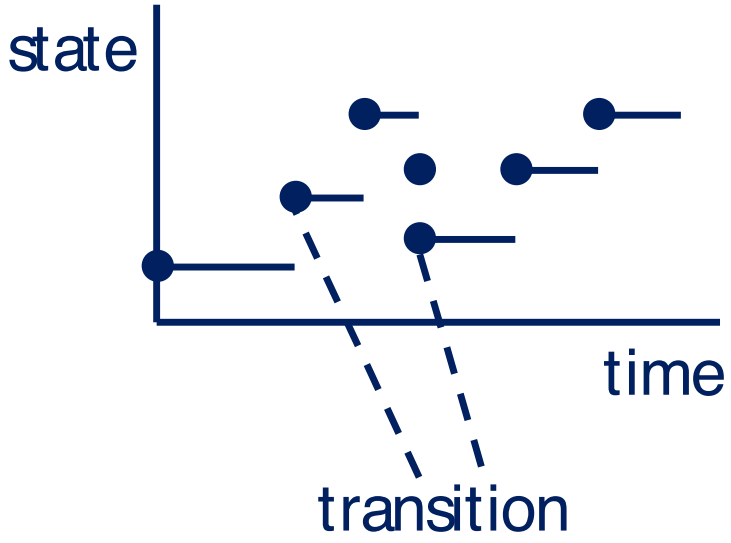
- **M1: Process Mining Overview, Positioning & Preliminaries (Event data & Process Models)**
- M2: Process Discovery
- M3: Conformance Checking
- M4: Process Enhancement

# Disclaimer

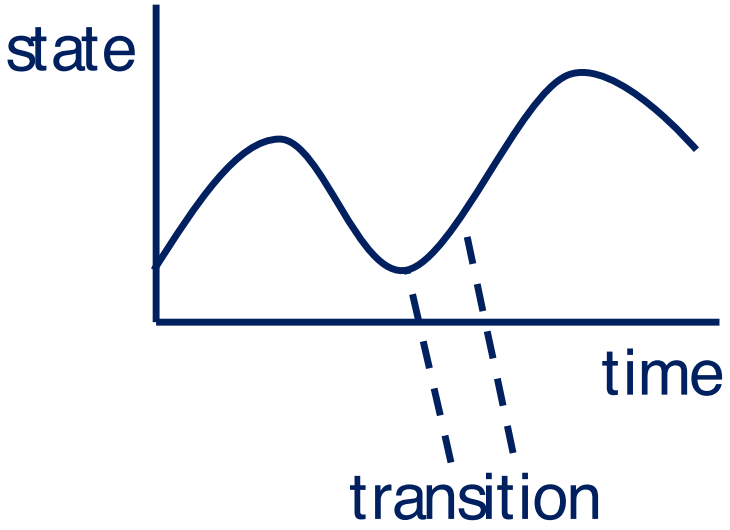
- Most of the material of this course is taken from my colleagues:
  - RWTH Aachen (Prof. Wil van der Aalst)
  - Humboldt University zu Berlin (Prof. Matthias Weidlich)
  - **Technishe Universiteit Eindhoven (Prof. Boudewijn van Dongen)**
  - University of Tartu (Prof. Marlon Dumas)
  - University of Melbourne (Prof. Marcello La Rosa)
  - Technical University of Denmark (Prof. Andrea Burattin)
- Hence, this material is only provided for your learning, please do not share nor publish

# Dynamic Systems: discrete vs continuous

## Discrete

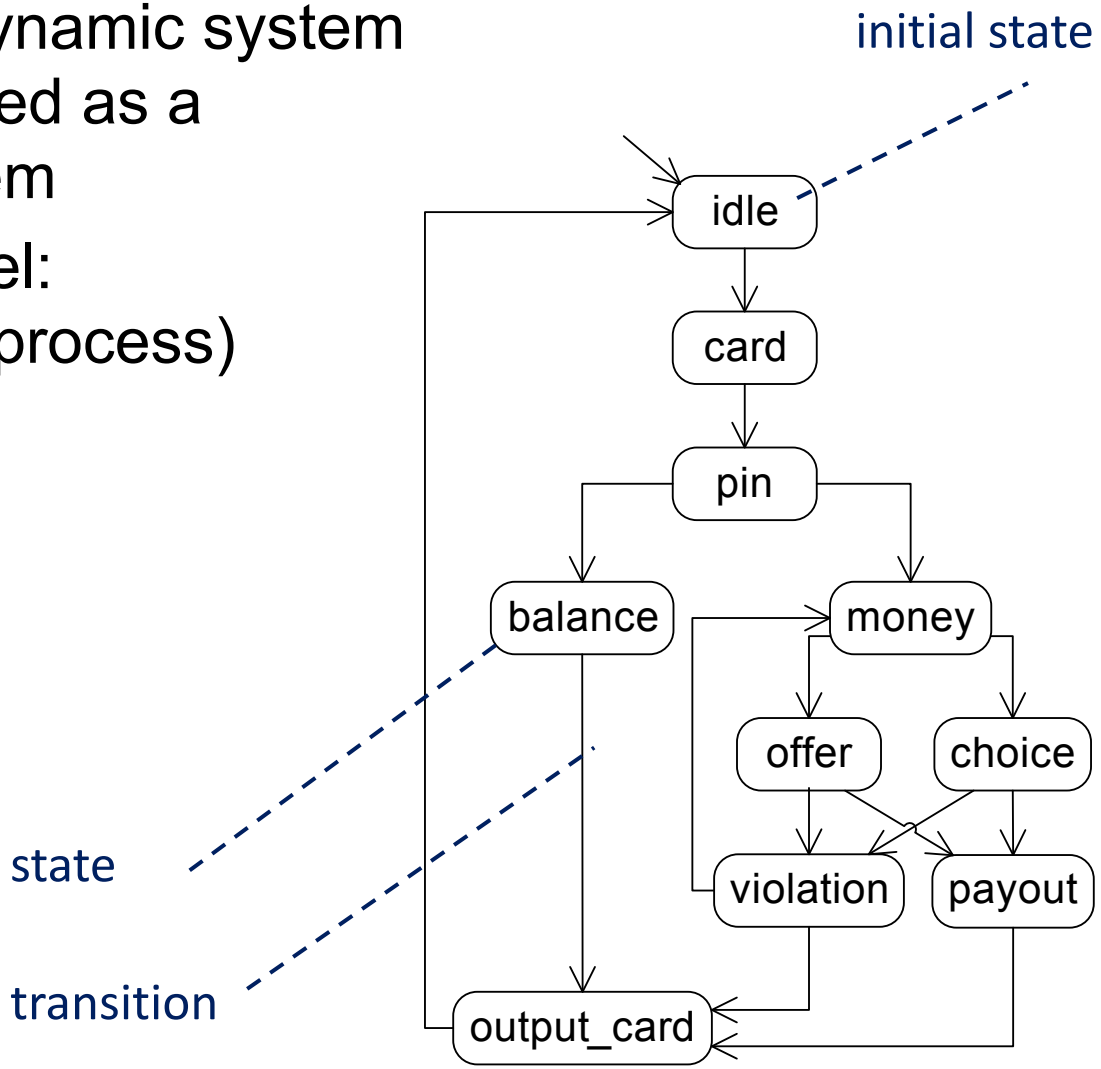



## Continuous



# Transition Systems

- Any discrete dynamic system can be described as a transition system
- Low level model: "Mother of all (process) models



# Example system: IEEE1394 (Firewire)

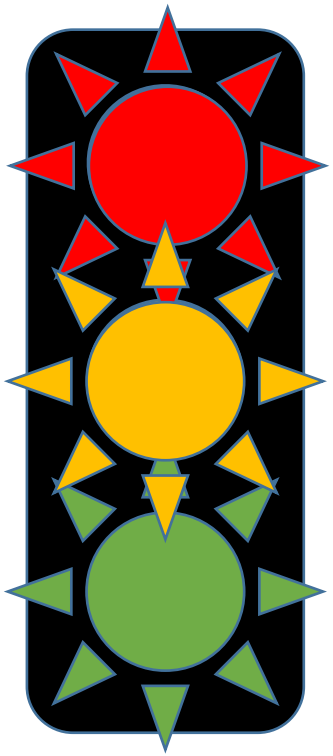
- Commucation between two nodes over a bus
- 25000 states
- Possible deadlocks!

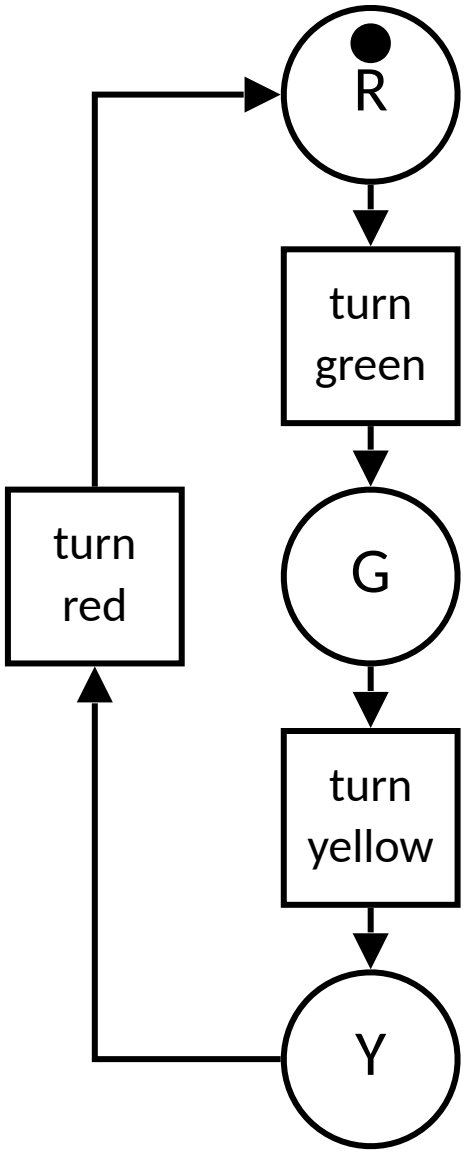
# Petri Nets

Petri nets provide a compact way to describe (potentially infinite) transition systems as a directed graph consisting of places and transitions, where:

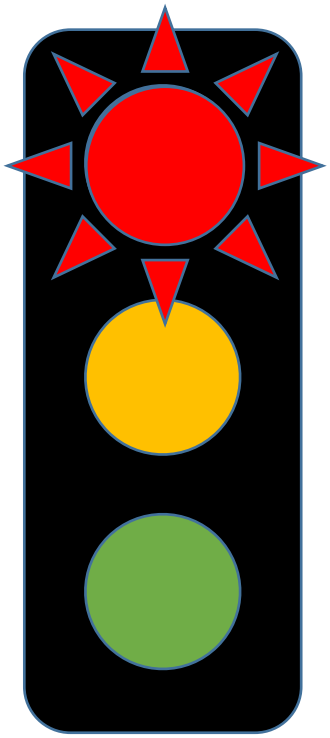
- Transitions are represented as boxes
- Places are represented as circles
- Places can contain 0 or more tokens
- The states of a transition system is defined as the distribution of tokens over places
- The Firing Rule governs the state changes

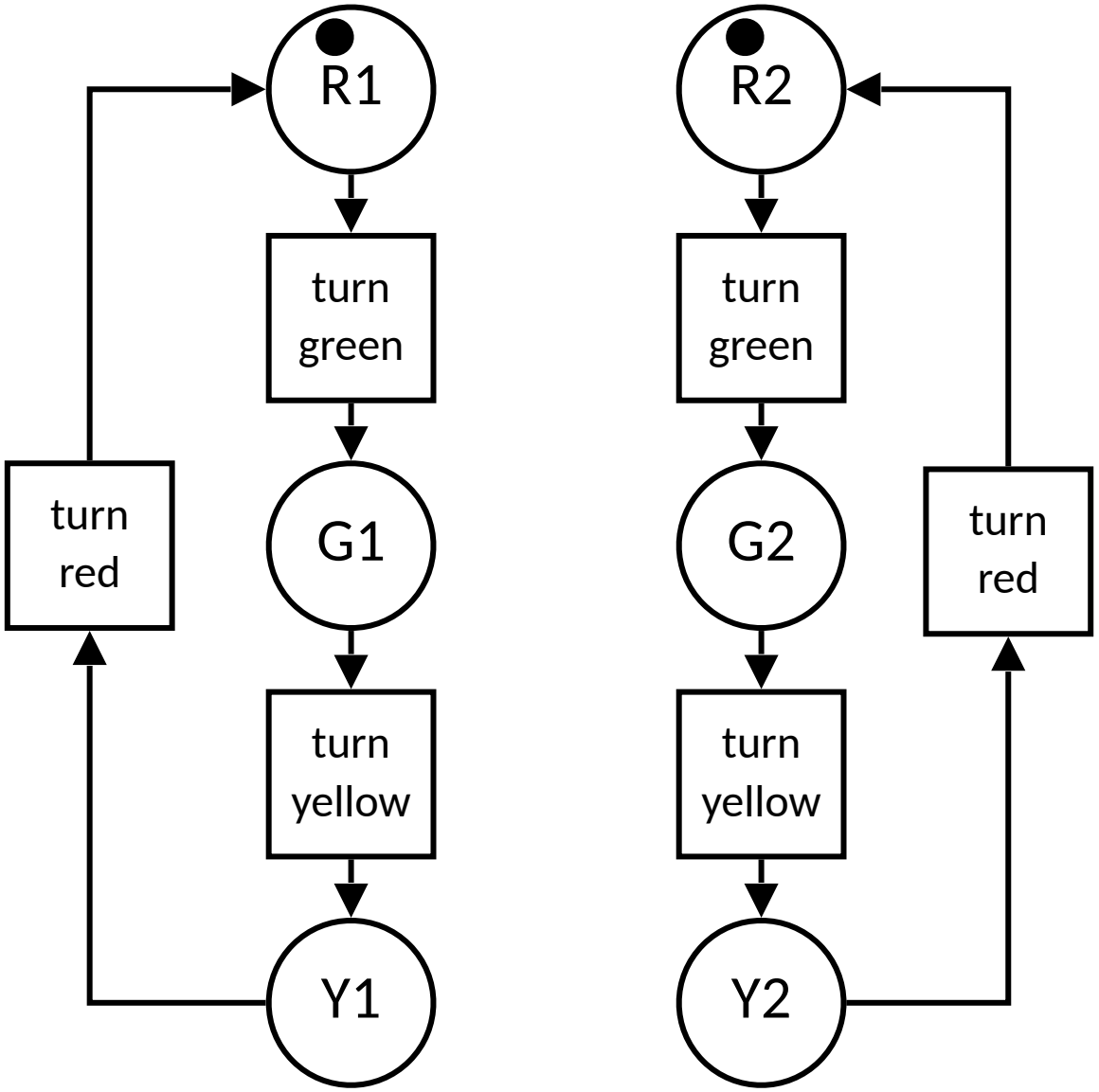
# Example: Single traffic light

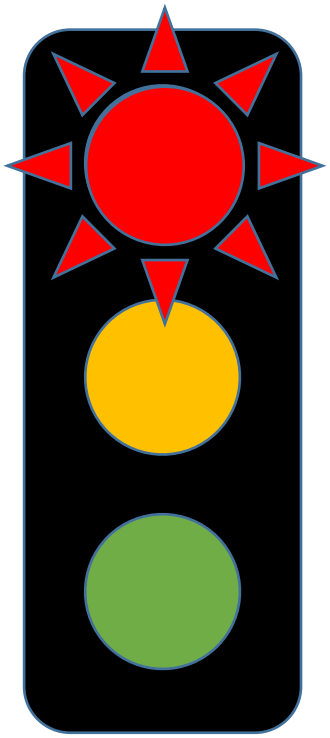




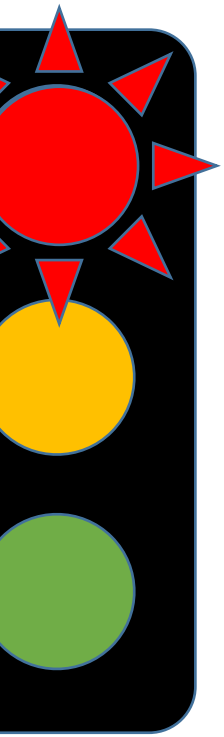
# Example: Two traffic lights

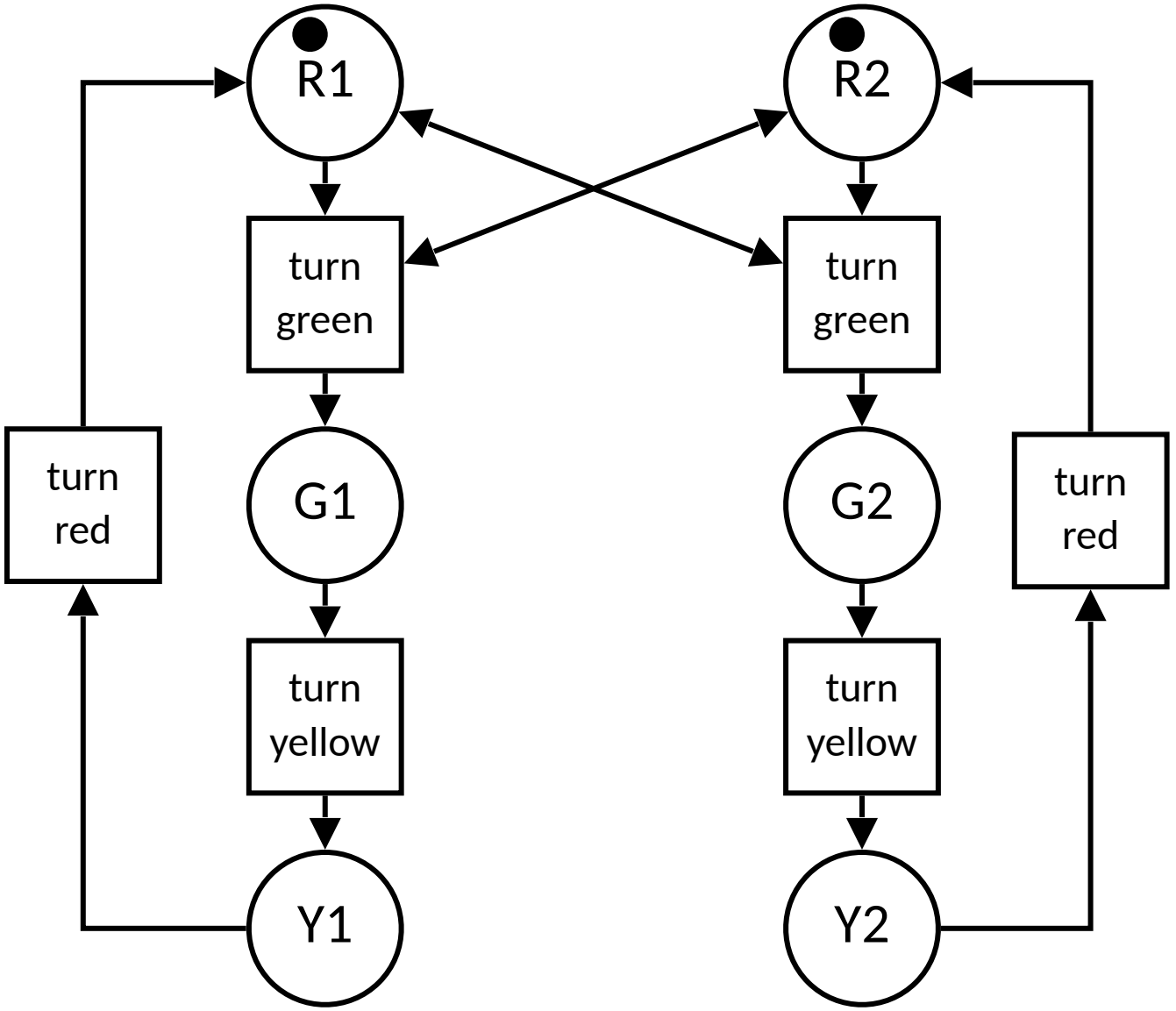


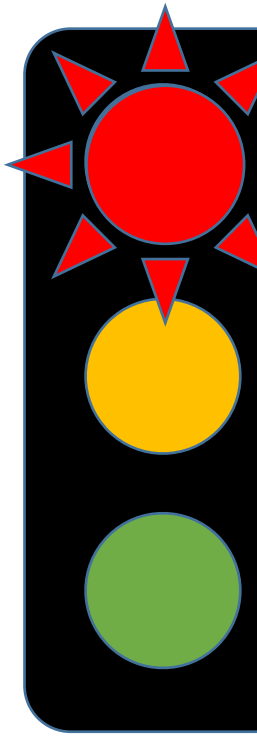


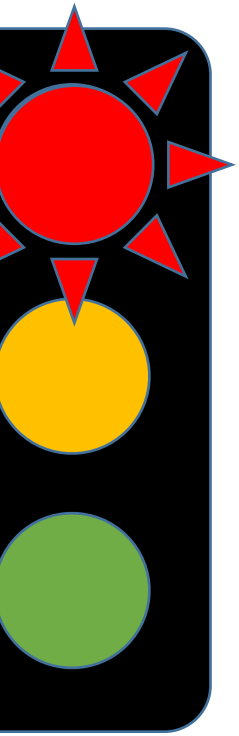
# Example: Two traffic lights

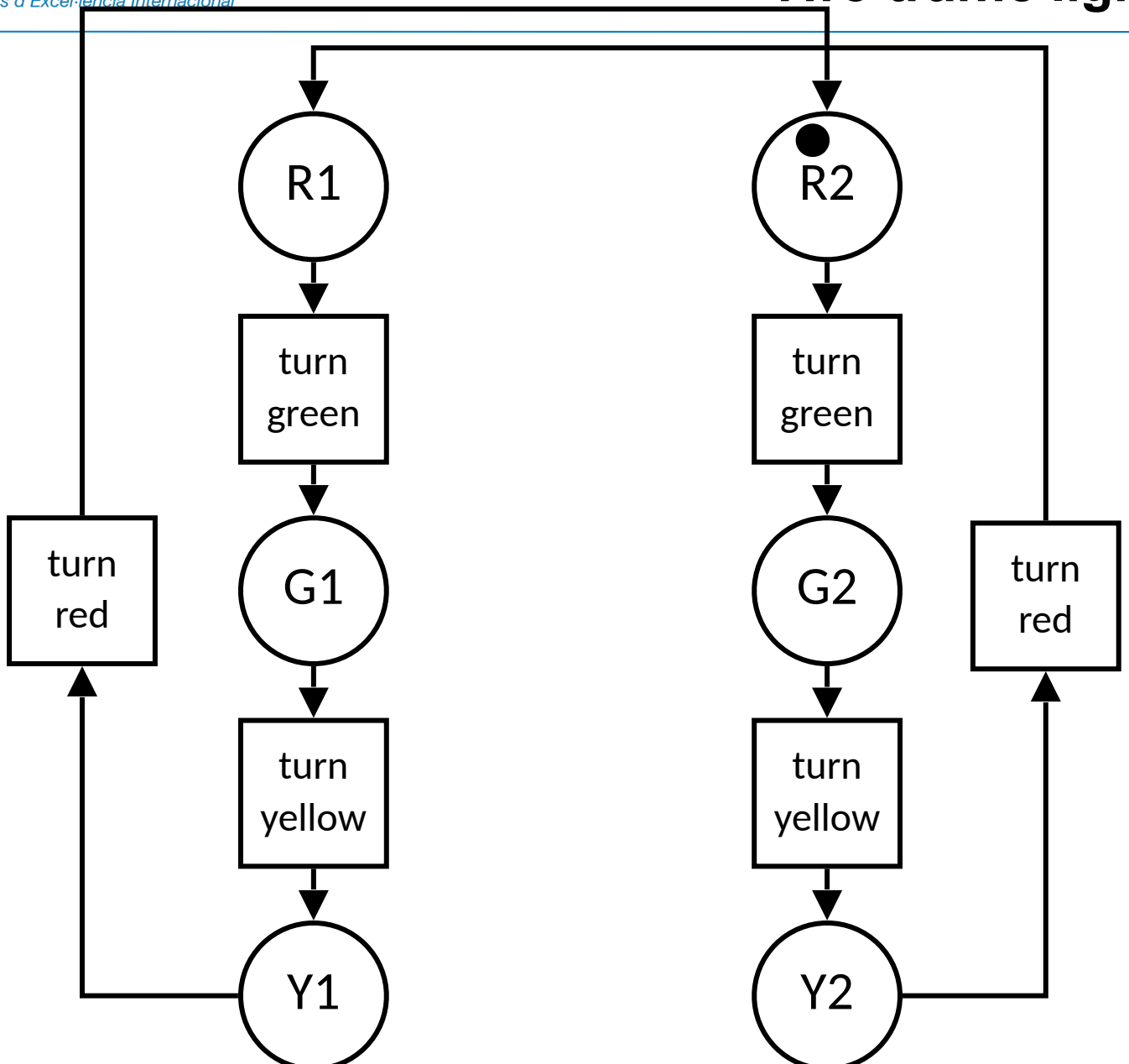


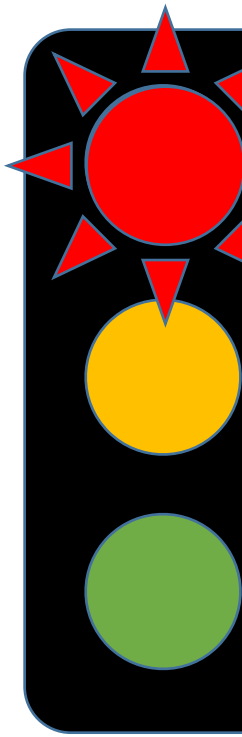


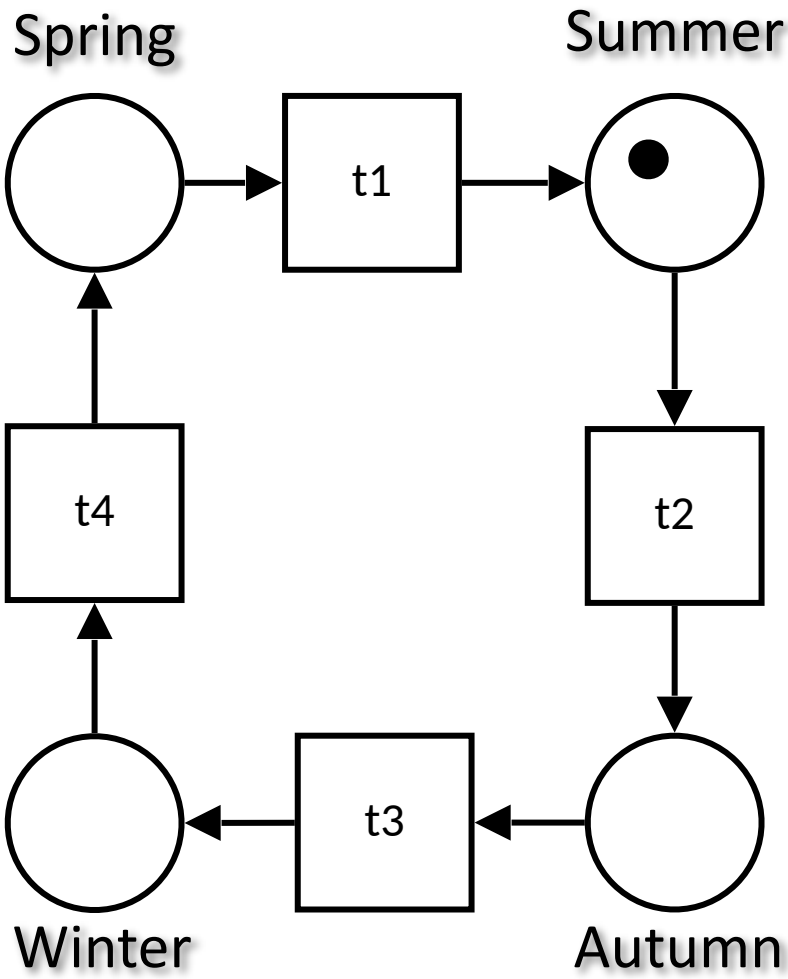



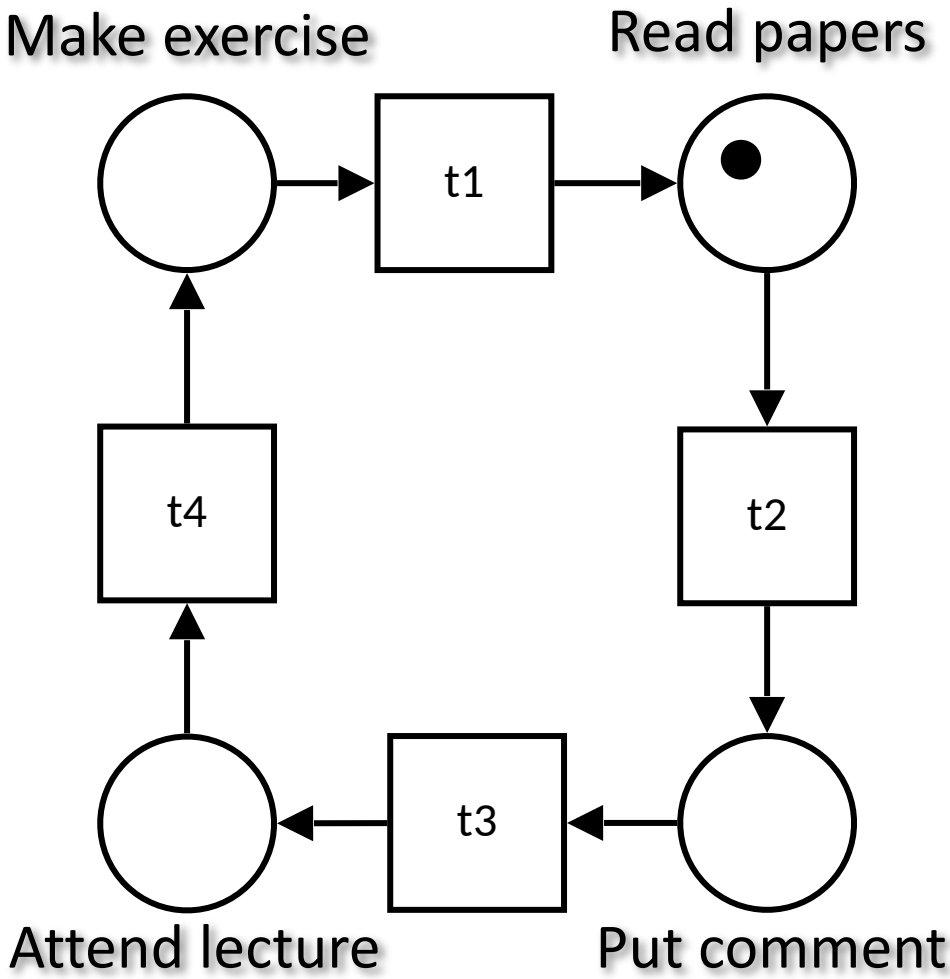
# Example: Two traffic lights








A model is an *abstraction* of reality

# Many systems can be modelled the same way!

# It's important that models are correct!

# Why Petri nets

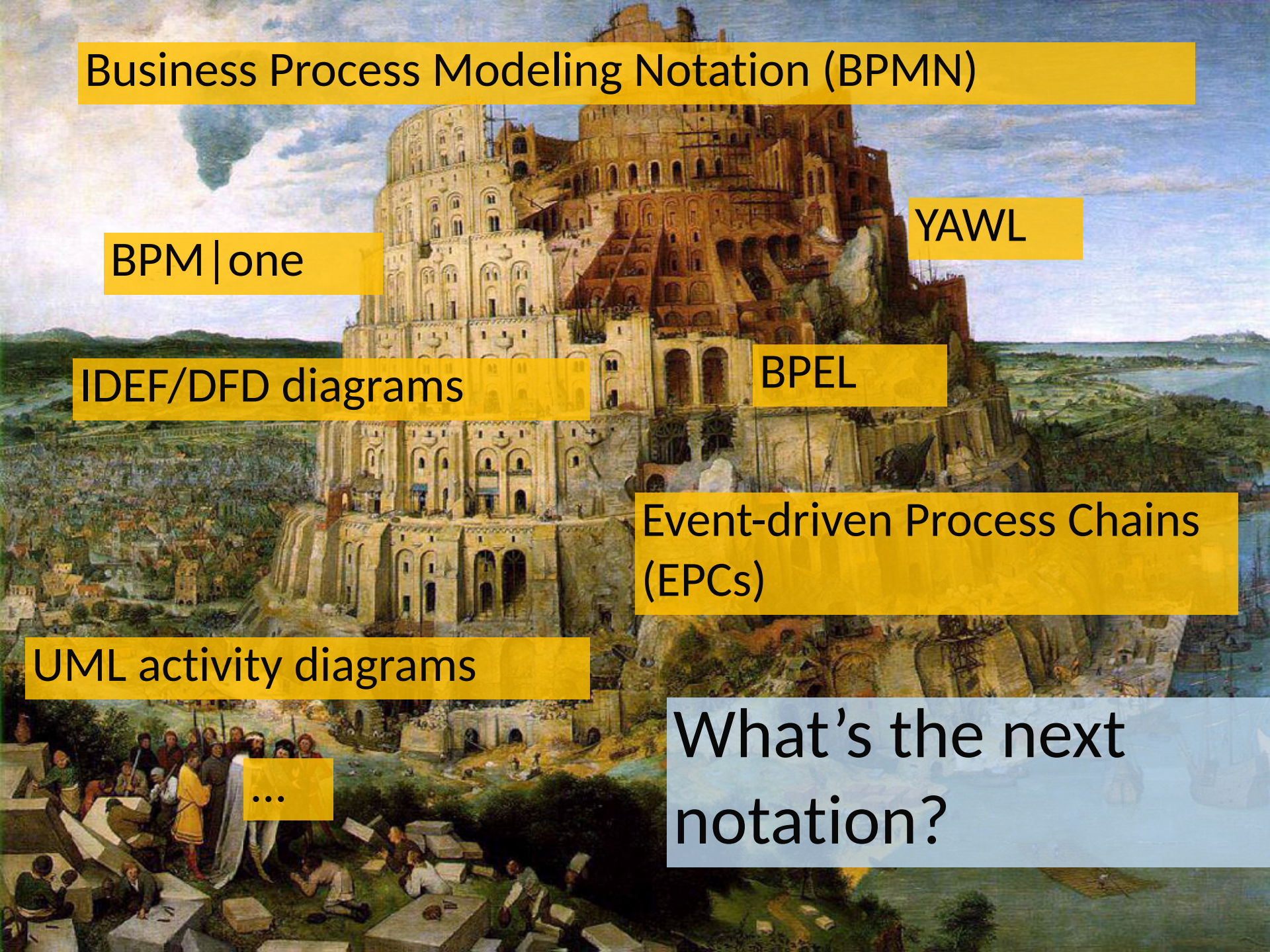
# The "Business Process Modelling " chaos: Several tools with different process languages

| Business process modeling tools | | | |
|---|---|---|---|
| Protos | ARIS | BPM\|one | IBM WebSphere |

| Model-based analysis tools | | | | | |
|---|---|---|---|---|---|
| Simula | Arena | IBM WebSphere | ProM | Wofla | CPN IDE |

| Business process enactment tools | | | |
|---|---|---|---|
| BPM\|one | SAP ERP | Oracle's JD Edwards | IBM WebSphere |

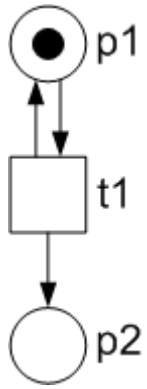
| Tools for analyzing running business process | | | | |
|---|---|---|---|---|
| ARIS | BPM\|one | IBM WebSphere | ProM | Fluxicon's Disco |


Business Process Modeling Notation (BPMN)
YAWL
BPM|one
BPEL
IDEF/DFD diagrams
Event-driven Process Chains (EPCs)
UML activity diagrams
…
What's the next notation?

# Why Petri Nets ?

## And not the most used notation in industry ...

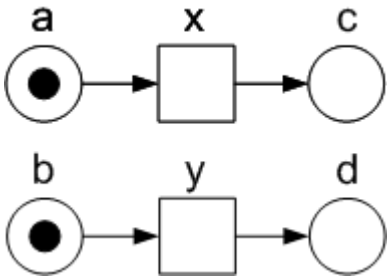
- All industrial languages/tools tend to come and go.
- Focus would be on syntactical elements rather than basic concepts.
- Often no formal semantics.
- Only few analysis techniques available.




Compactness

Goal:

- Teach foundational concepts of process modeling and analysis.




Concurrency, locality

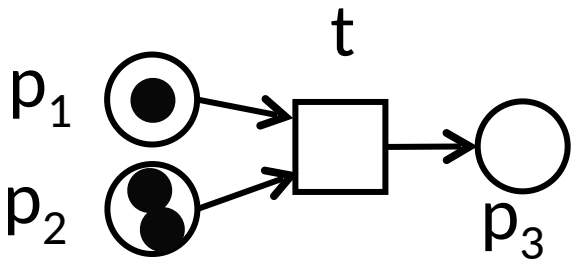
Tool support

# Petri Nets: Formalization

A Petri net is a triple $(P, T, F)$, where

- $P$ is a finite set of places,
- $T$ is a finite set of transitions,
- $F \subseteq (P \times T) \cup (T \times P)$ is a flow relation.

Any diagram can be mapped onto such a triple and vice versa.

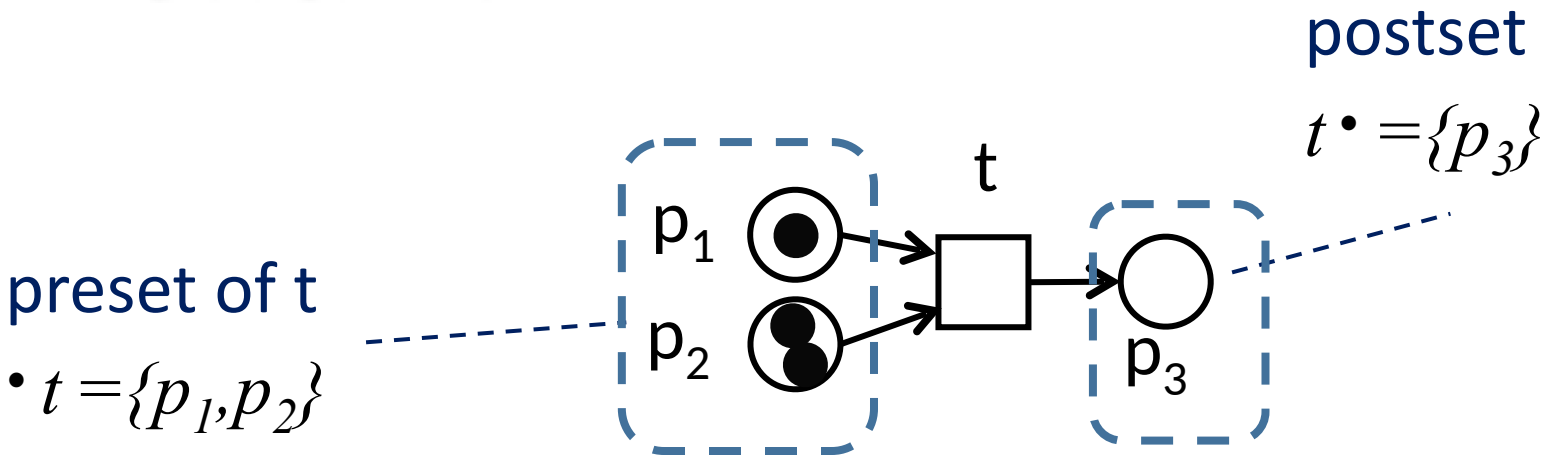



$$P = \{\text{p1, p2, p3}\}$$
$$T = \{\text{t}\}$$
$$F = \{(\text{p1, t}), (\text{p2, t}), (\text{t, p3})\}$$

# Petri Nets: Preset and Postset

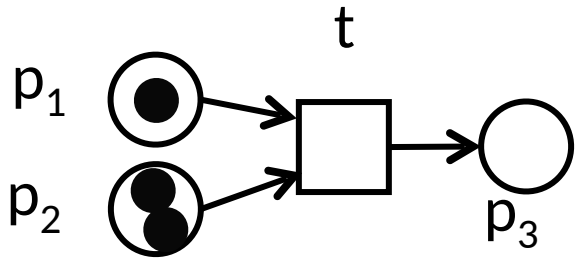
Intuition: Define input and output places in terms of the flow relation.
Let $t$ be a transition.

- Set ${}^\bullet t = \{p \mid (p, t) \in F\}$ defines all input places of $t$ (the preset of $t$)
- Set $t^\bullet = \{p \mid (t, p) \in F\}$ defines all output places of $t$ (the postset of $t$)



# Petri Nets: Enabledness

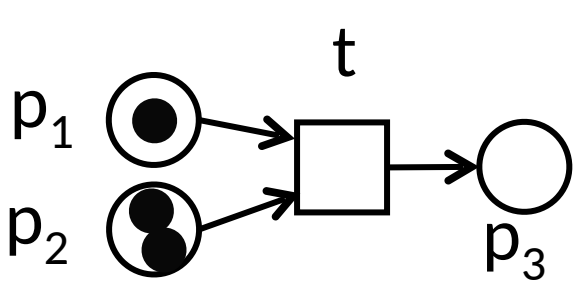
In a Petri net $(P, T, F)$, a transition $t \in T$ is enabled at marking $m: P \rightarrow \mathbb{N}$ if for all $p \in {}^\bullet t$, $m(p) > 0$.




*t* is enabled at marking *m*

# Petri Nets: Firing Rule

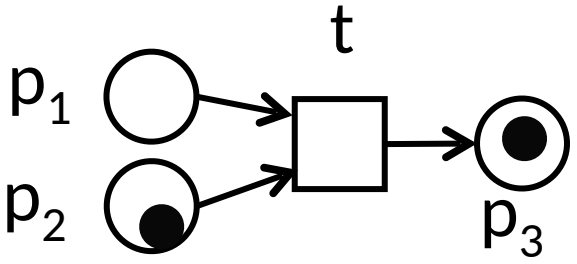
- For a Petri net $(P, T, F)$, firing transition $t \in T$ at a marking $m_1$ leads to a new marking $m_2$ such that for any $p \in P$
  - $m_2(p) = m_1(p) - 1$ if $p \in {\bullet t} \setminus t^{\bullet}$
  - $m_2(p) = m_1(p) + 1$ if $p \in t^{\bullet} \setminus {\bullet t}$
  - $m_2(p) = m_1(p)$ otherwise




$m_1 = [p_1, p_2^2]$

firing t yields:

$m_2(p_1) = m_1(p_1) - 1$
$m_2(p_2) = m_1(p_2) - 1$
$m_2(p_3) = m_1(p_3) + 1$




$m_2 = [p_2, p_3]$

# What's a process ?

*Any time two or more activities are performed in the context of a case, fundamental principles of processes apply (e.g. Petri net theory)*

Activities in processes are related. They have a logical order in which they have to be executed.

Cases are the 'objects' in a process that change over time. For example an insurance claim, an offence registration or a loan application

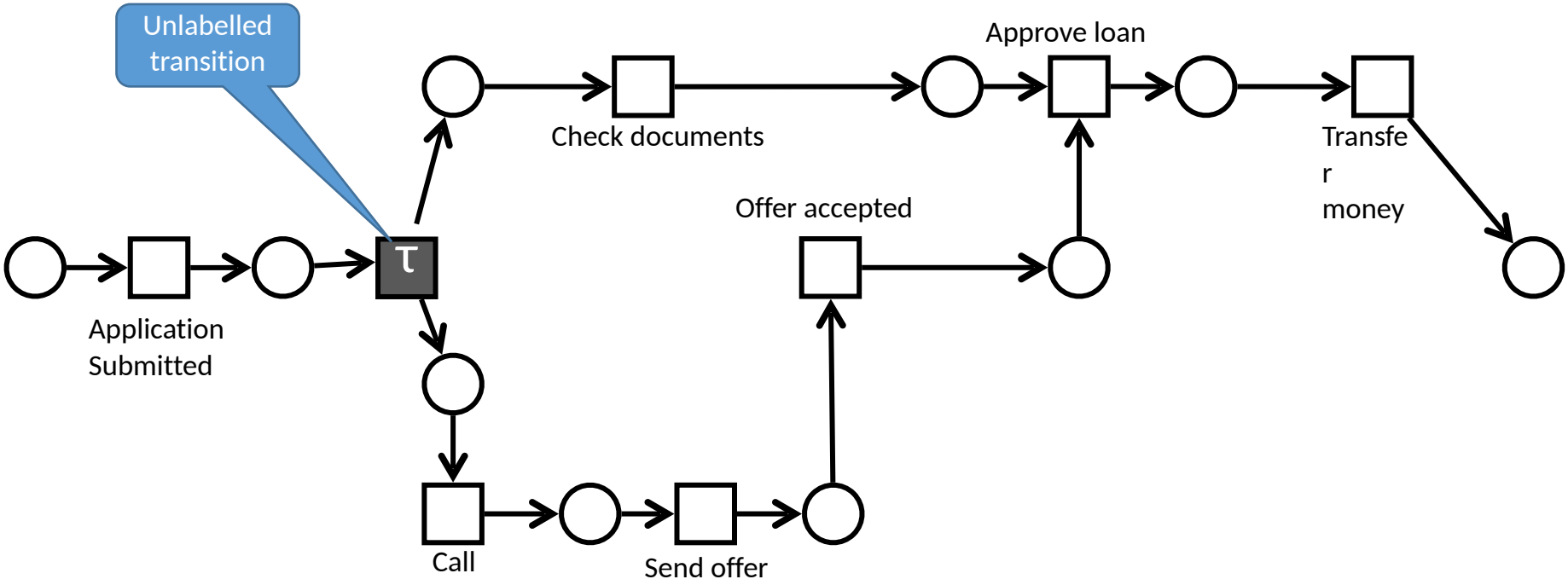
# Example: BPI 2012 (simplified to fit slides)

Process description from the company's website:

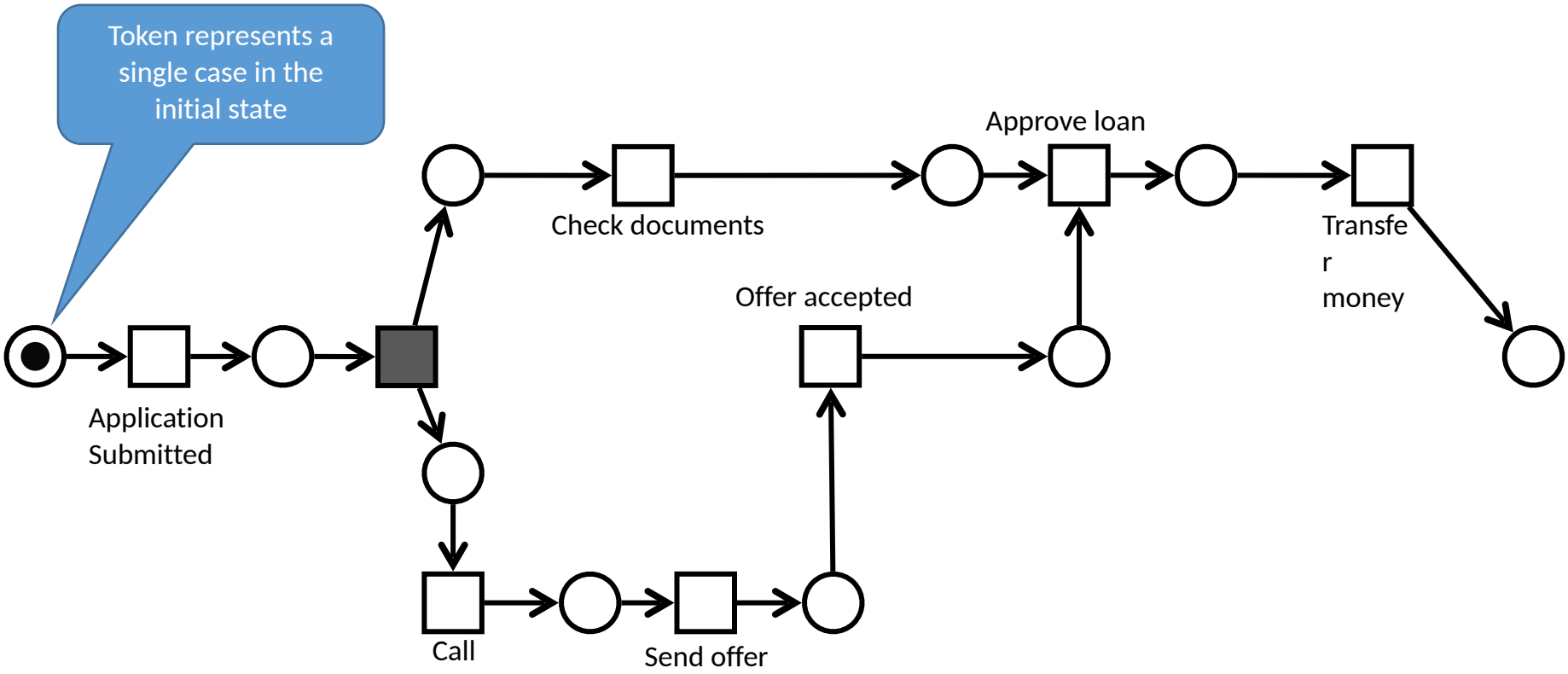
- *The nice thing about taking a loan from us is that you can arrange everything online. A call agent will then contact you to go through your application, to judge the application and to send you an offer for signing.*

- *After your approval of the offer and a final check by our customer center if all submitted documents are correct, we transfer the money to your account.*

# Example: BPI 2012 (simplified to fit slides)



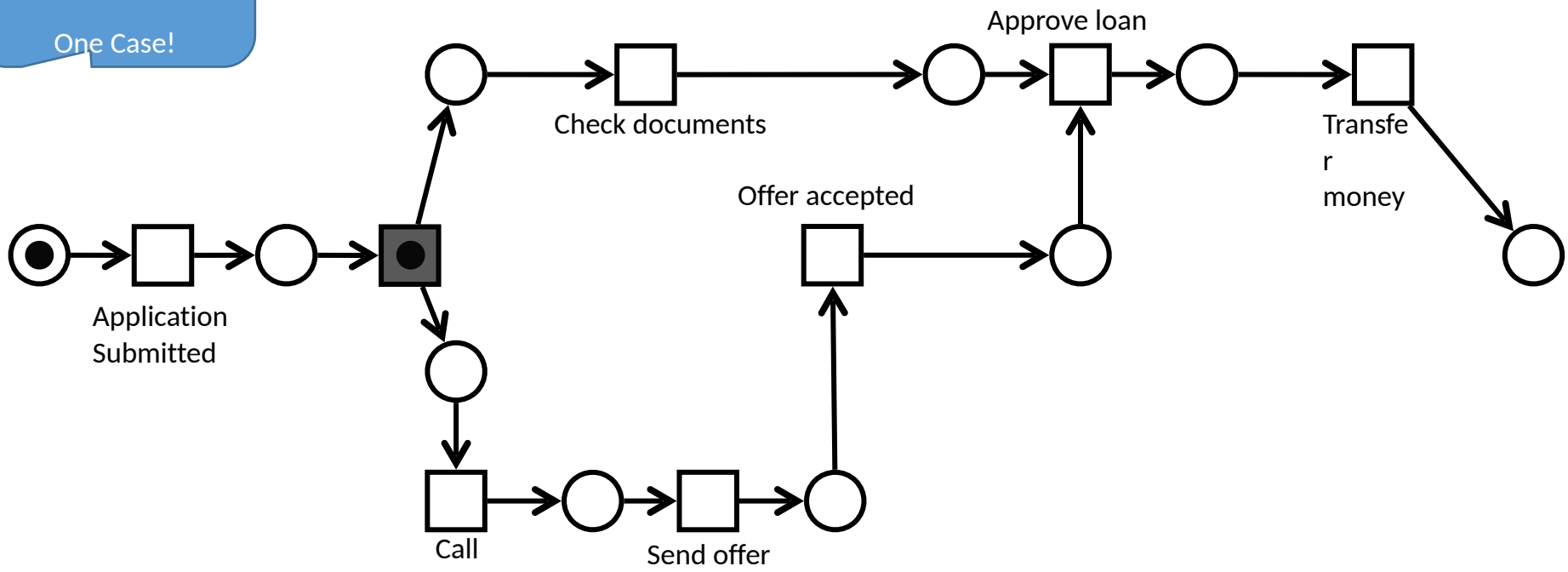
# Example: BPI 2012 (simplified to fit slides)



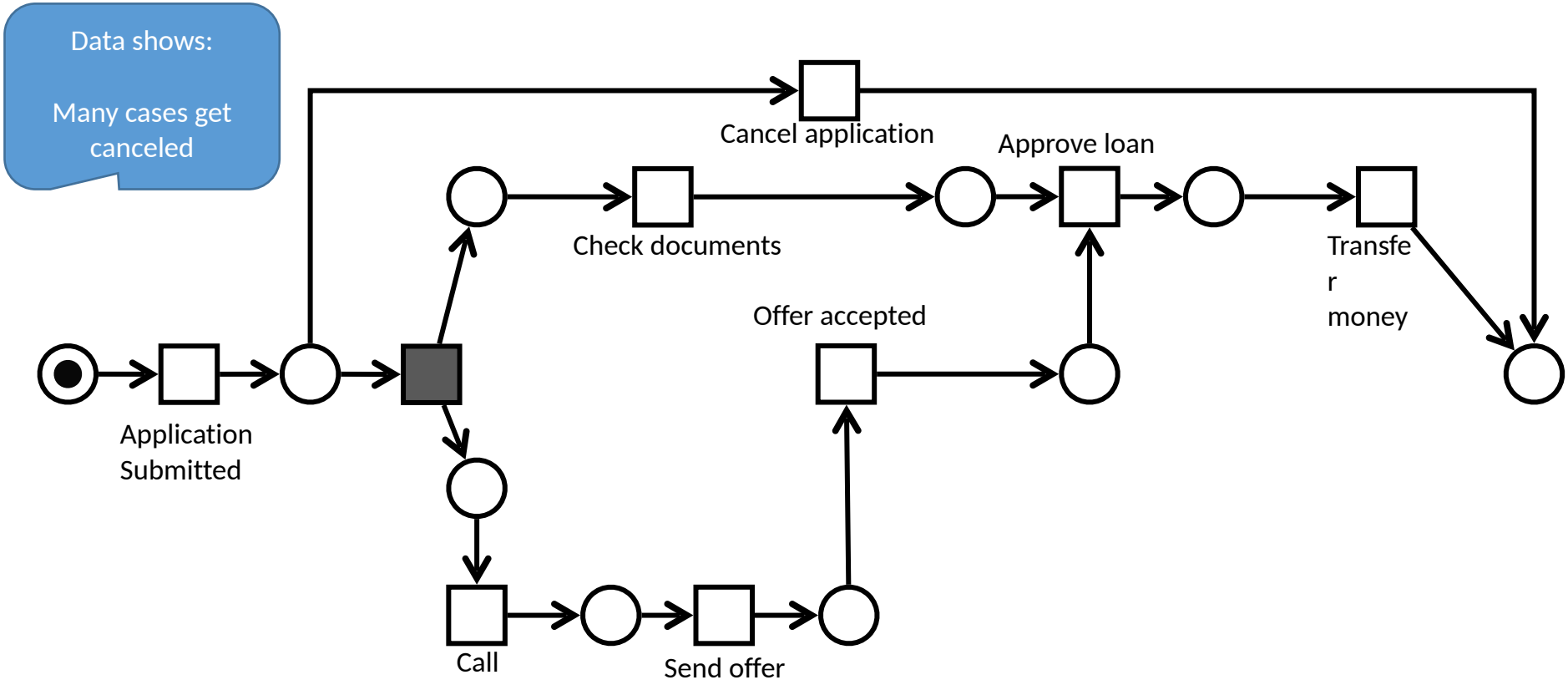
# Example: BPI 2012 (simplified to fit slides)

Happy-flow execution of this process:

One Case!

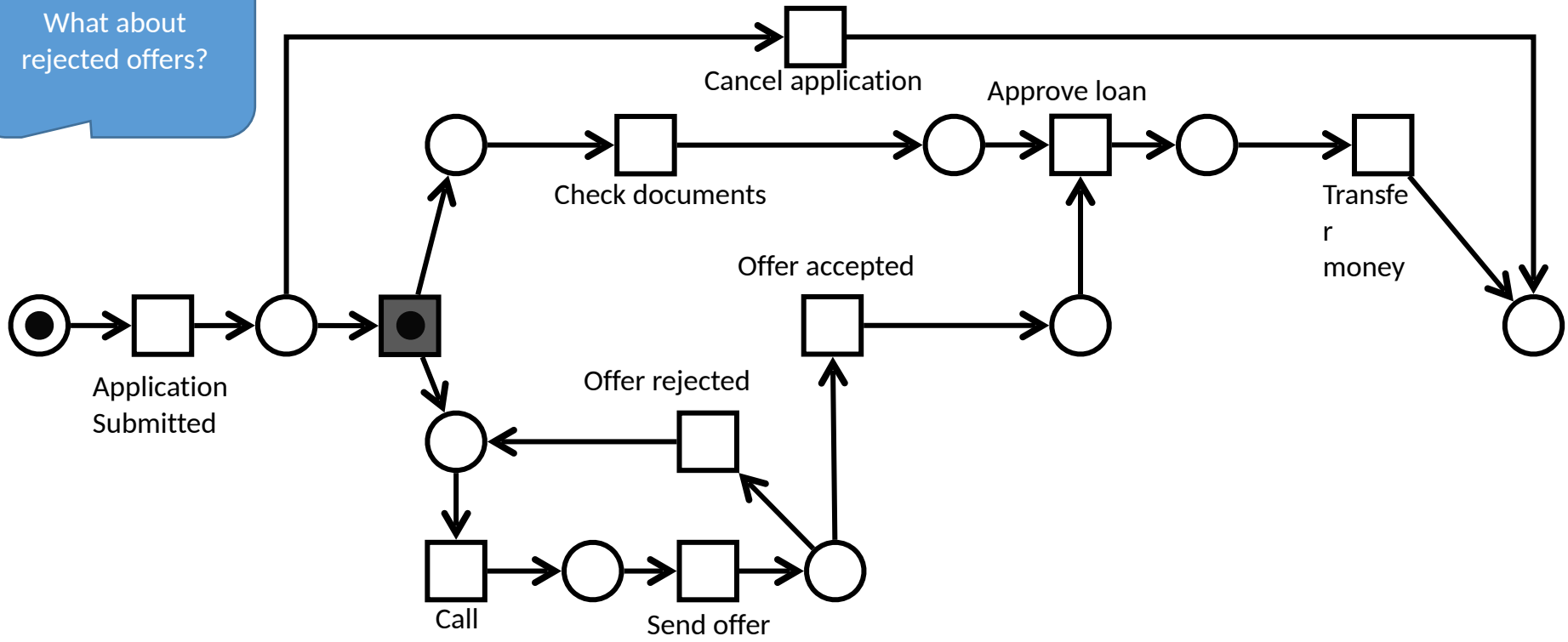


# Example: BPI 2012 (simplified to fit slides)

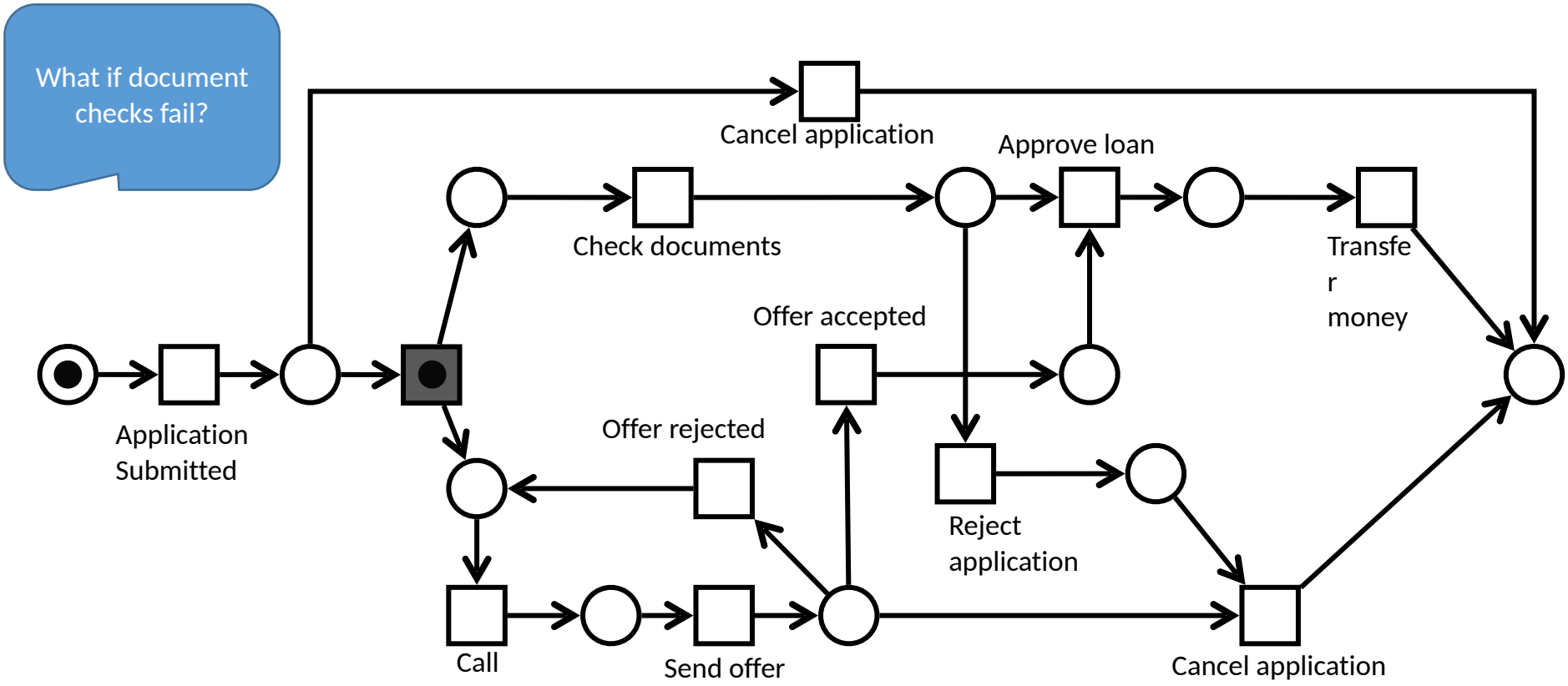


# Example: BPI 2012 (simplified to fit slides)

What about rejected offers?

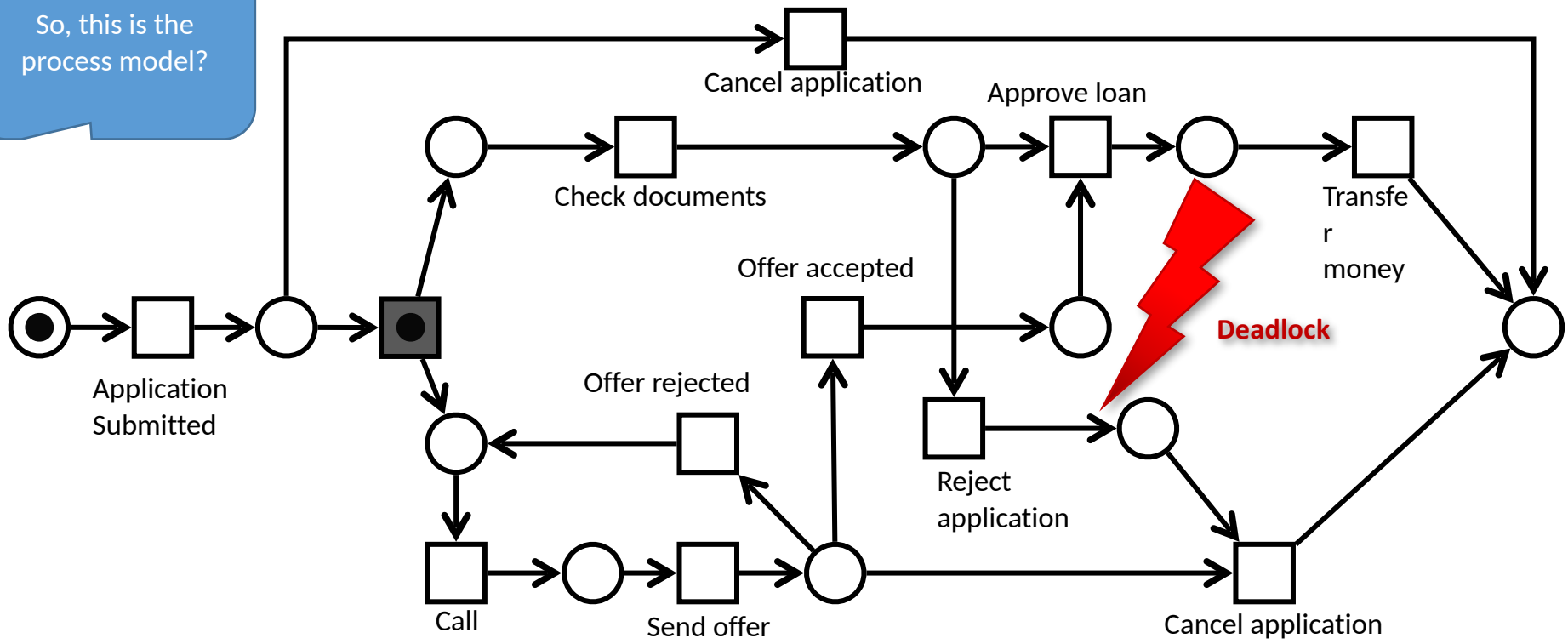


# Example: BPI 2012 (simplified to fit slides)

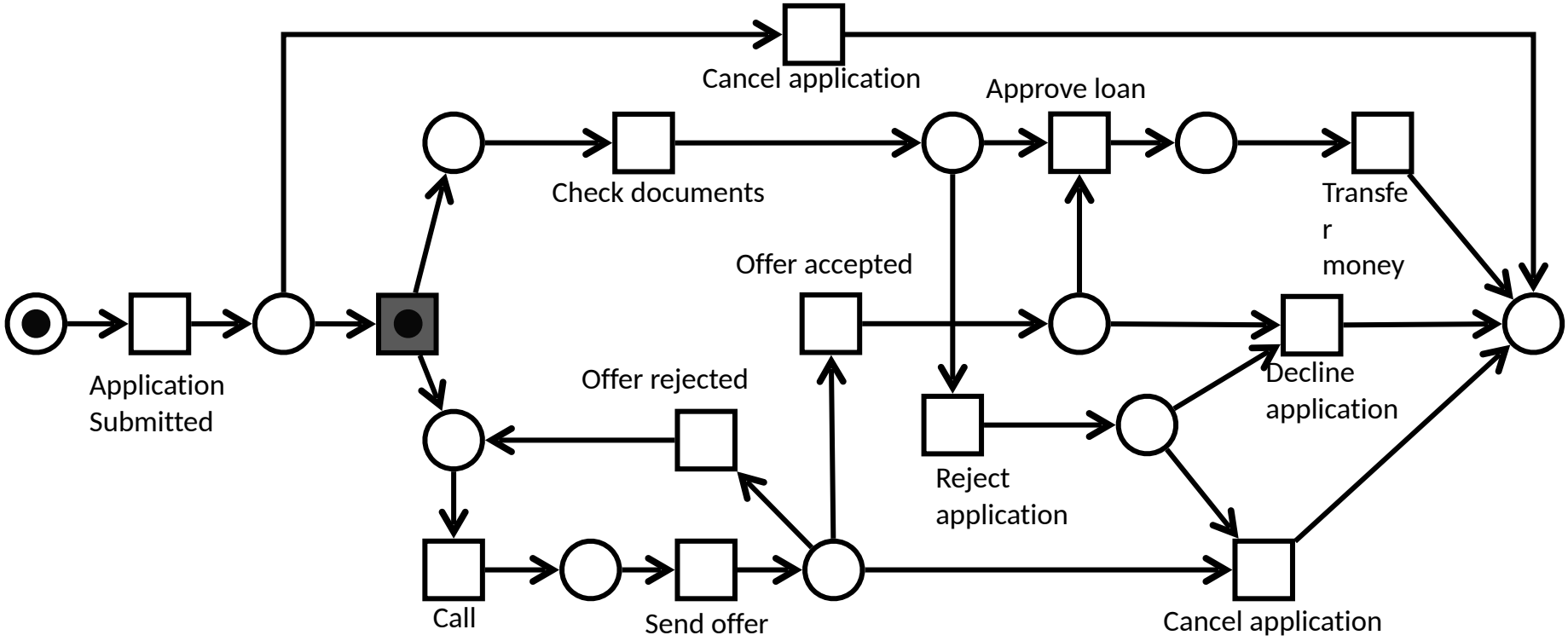


# Example: BPI 2012 (simplified to fit slides)

So, this is the process model?

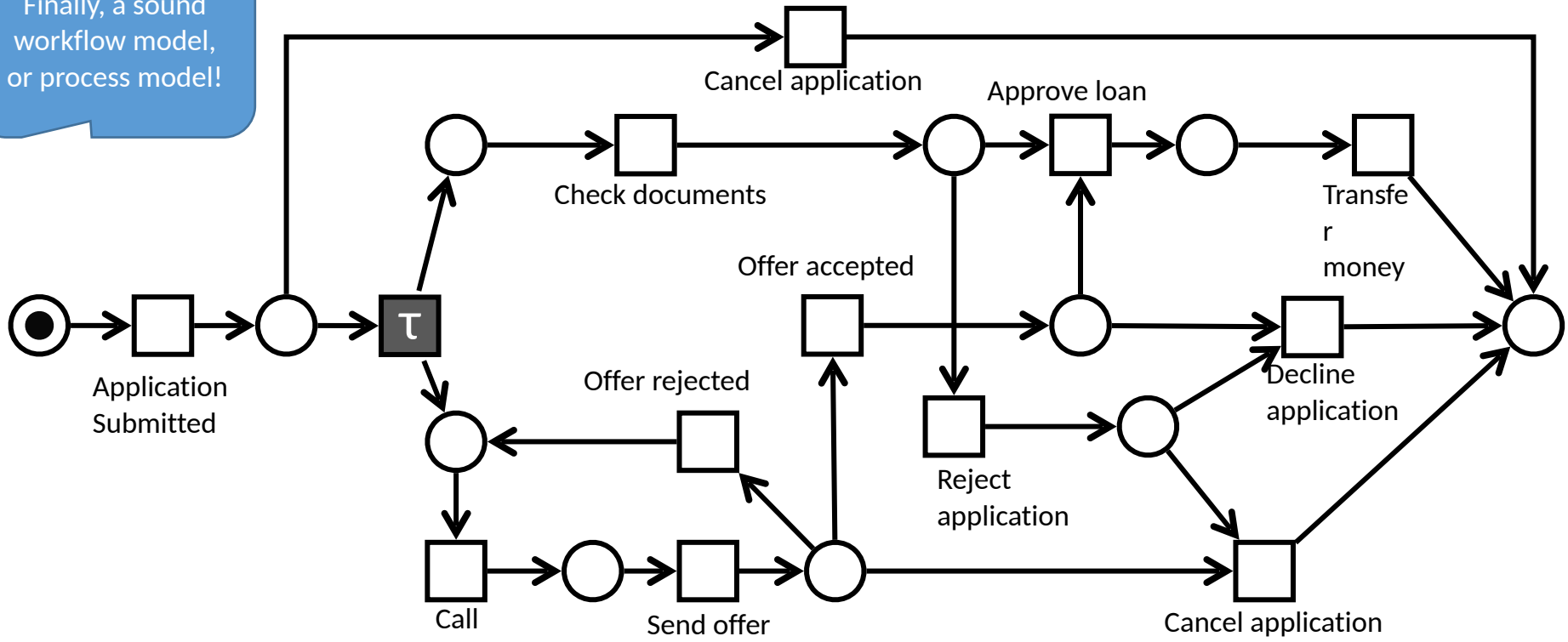


# Example: BPI 2012 (simplified to fit slides)



# Example: BPI 2012 (simplified to fit slides)

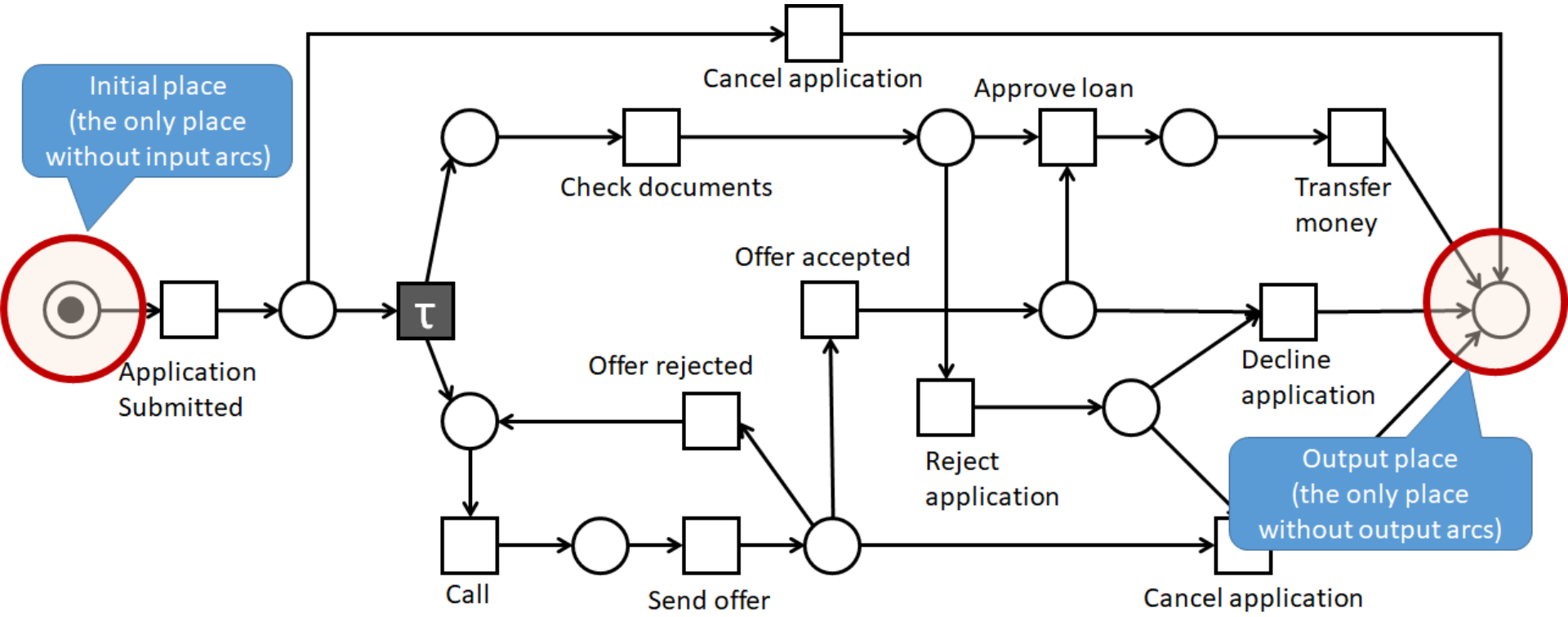
Finally, a sound workflow model, or process model!



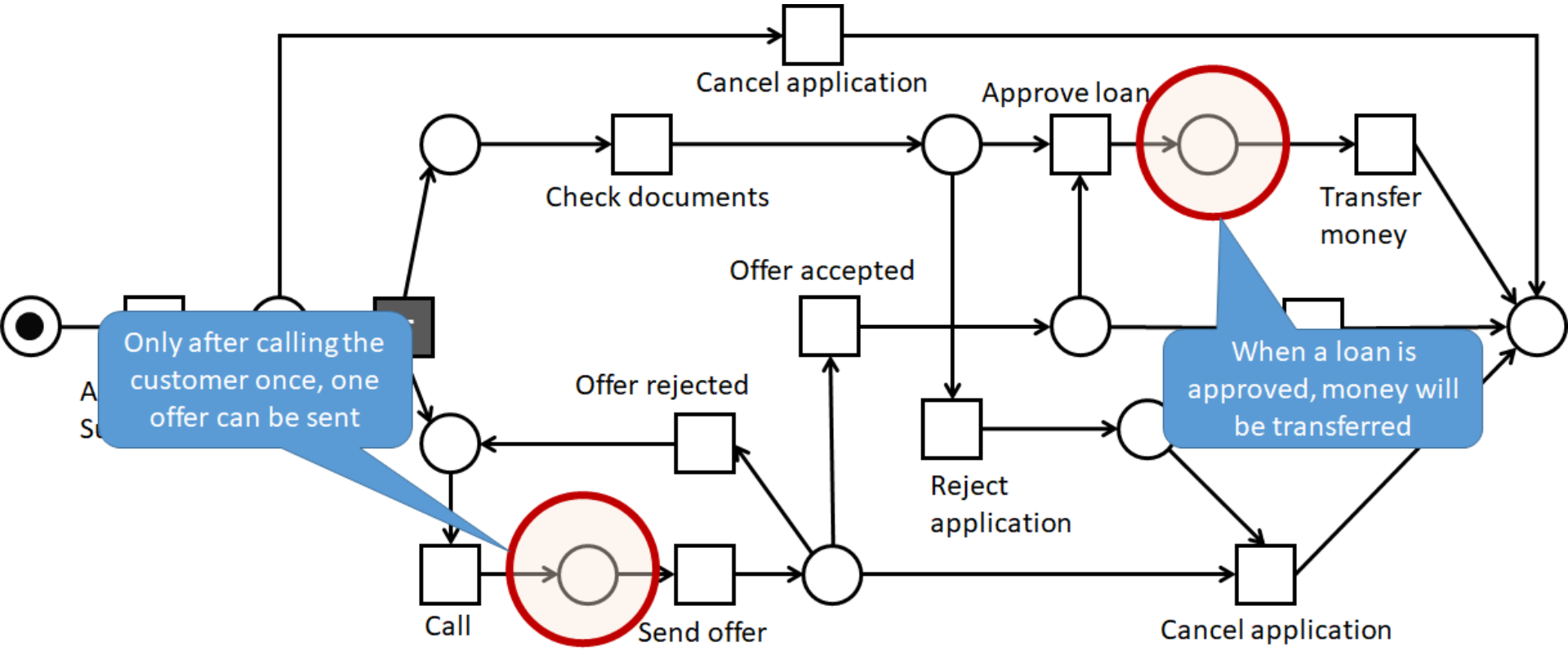
# Process Models vs. Petri Nets

| Process | Petri net |
| --- | --- |
| • Activity by resource | • Transition (-label, typically present tense) |
| • Decision by environment | • Transition (-label, typically past tense) |
| • Process Model | • Petri net explaining the flow of each case |
| • Case | • Token(s) in the net |
| • State of a case | • Marking of the Petri net |

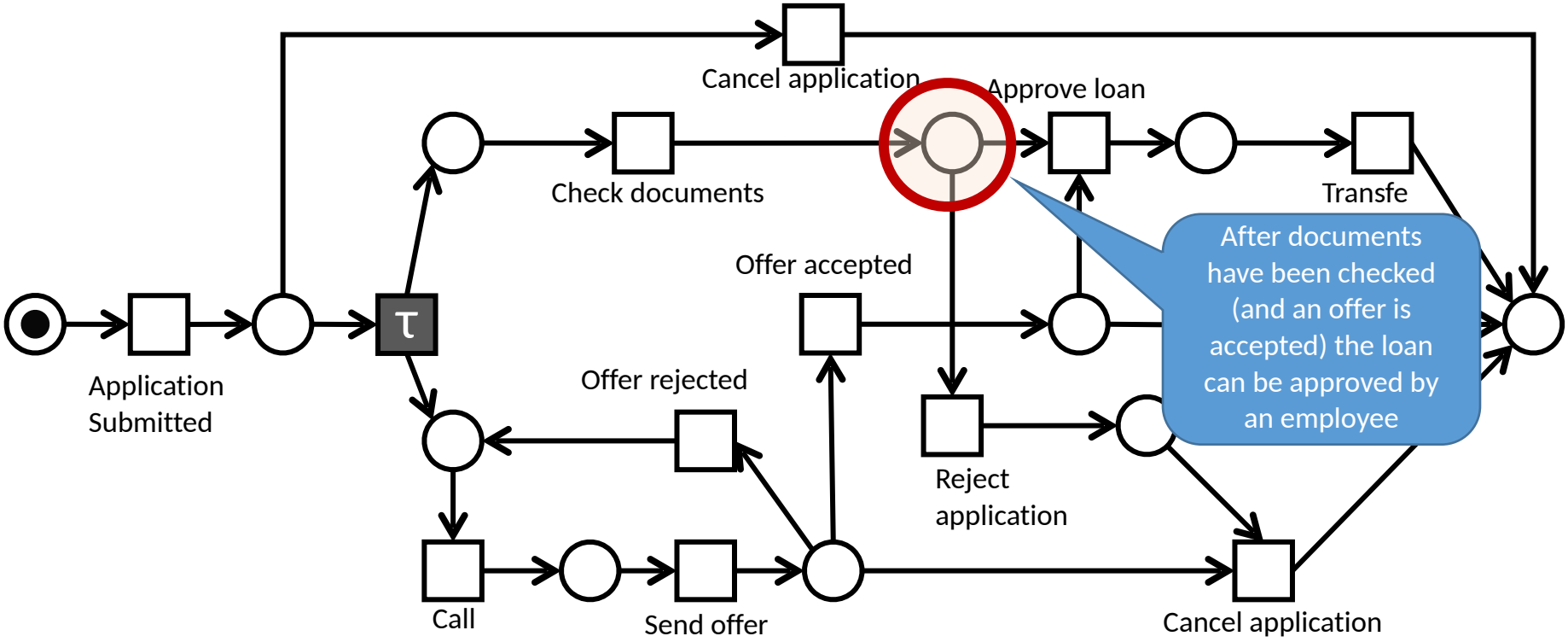
# Process: Proper begin and end



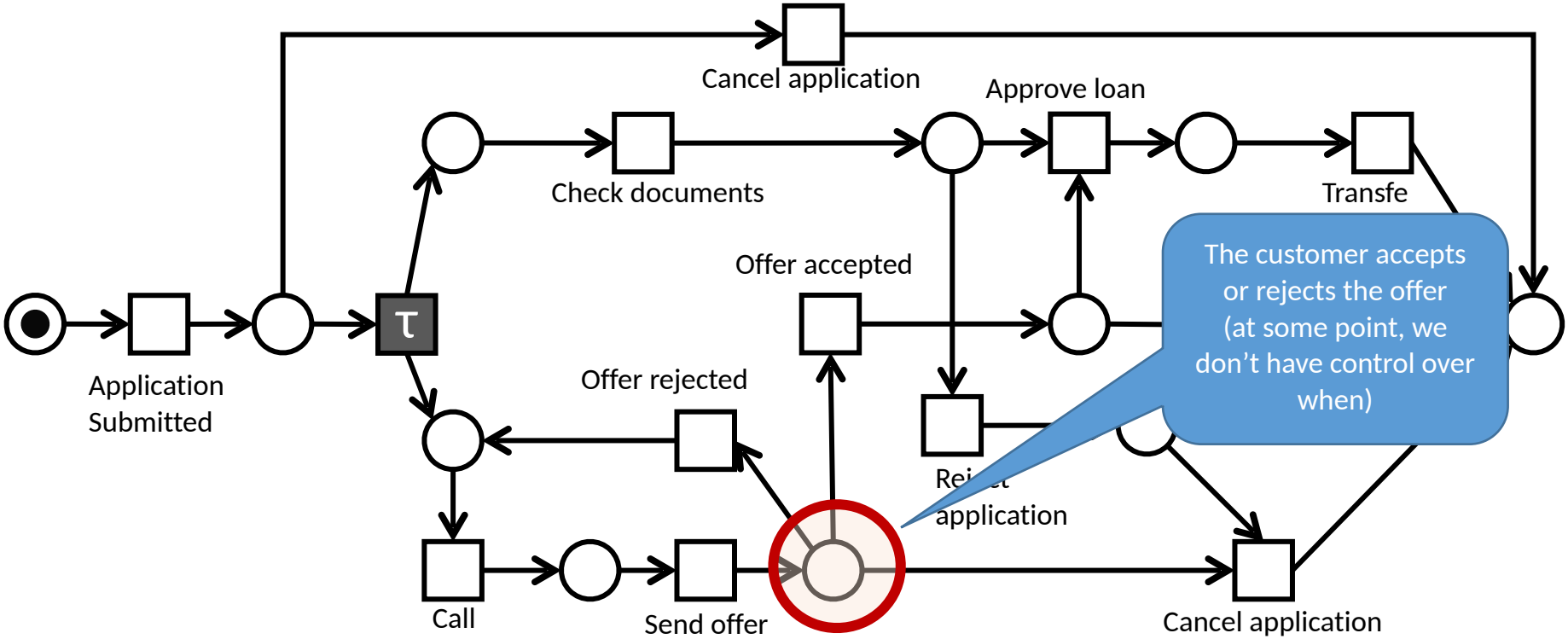
# Connecting places imply causal dependencies



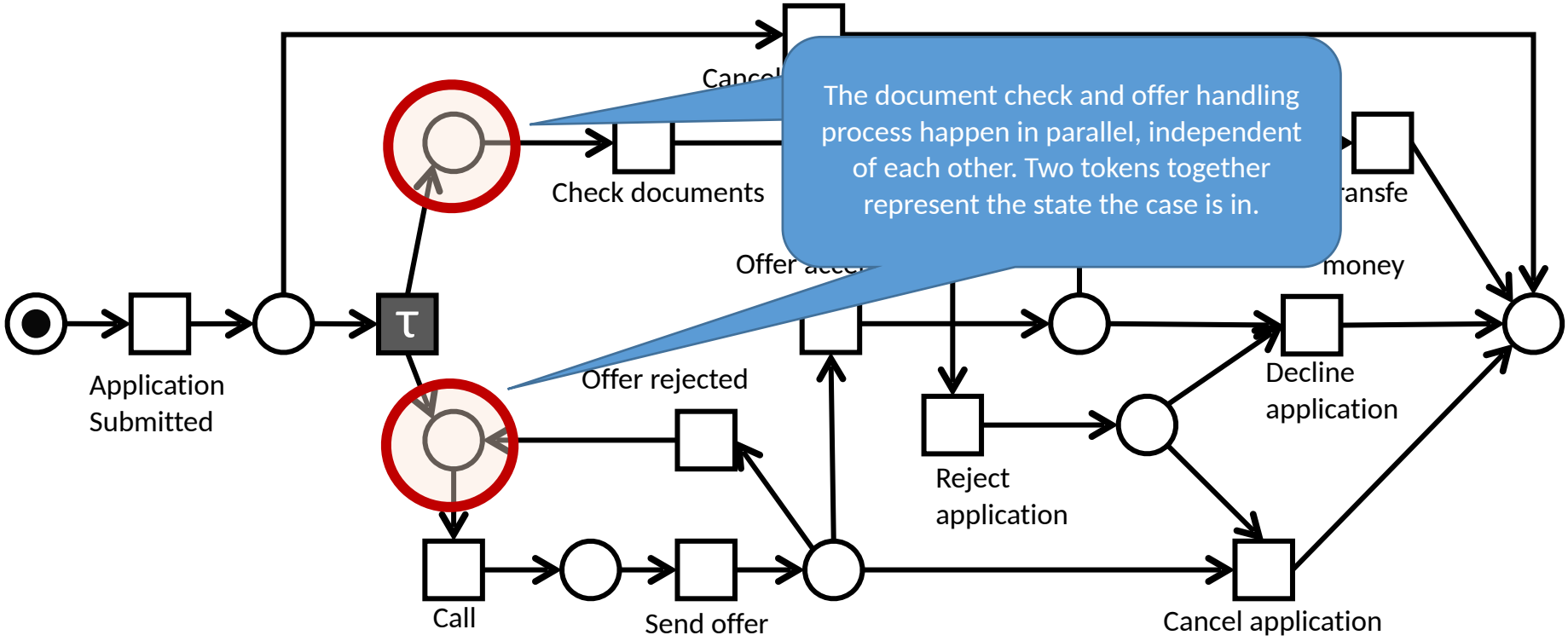
# Explicit Choices (internal to the process)



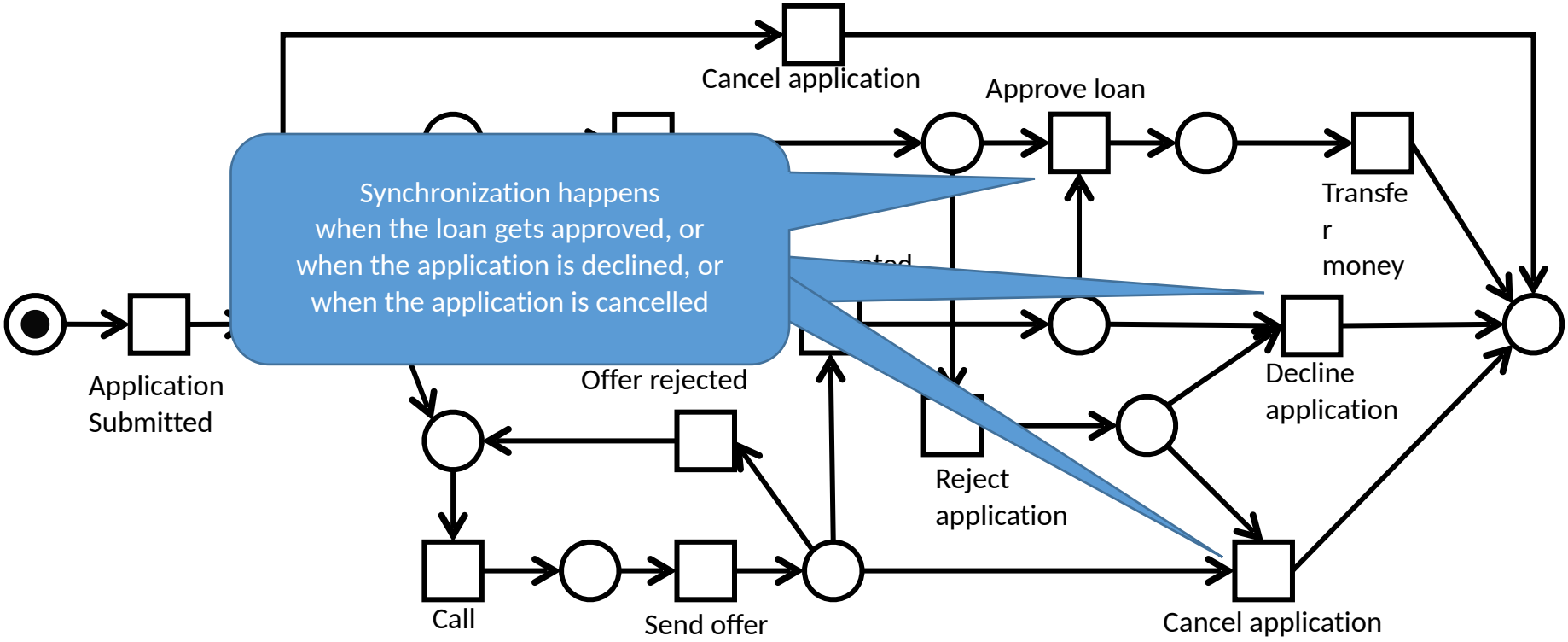
# Implicit Choices (external to the process)



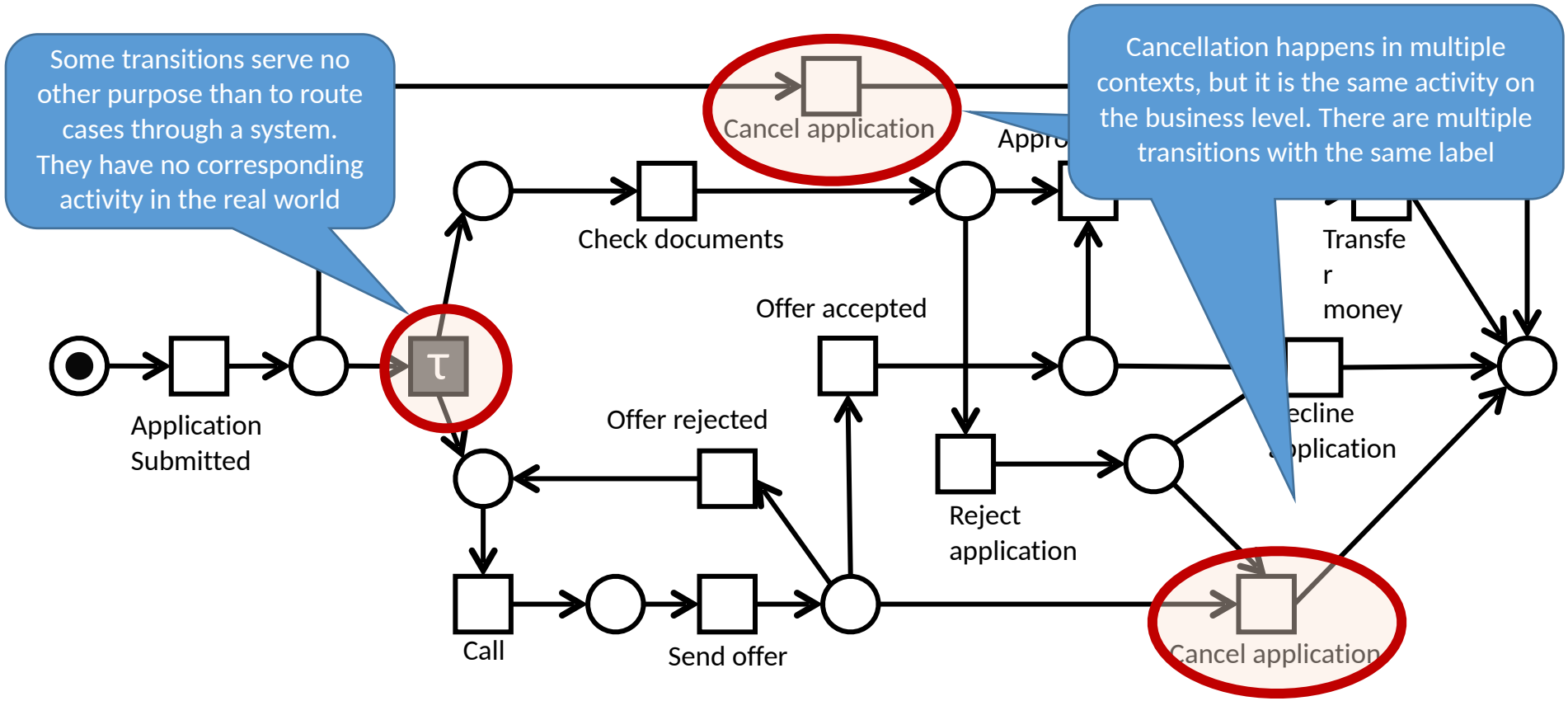
# Paralellism



# Paralellism



# Duplicate labels / 'routing' transitions

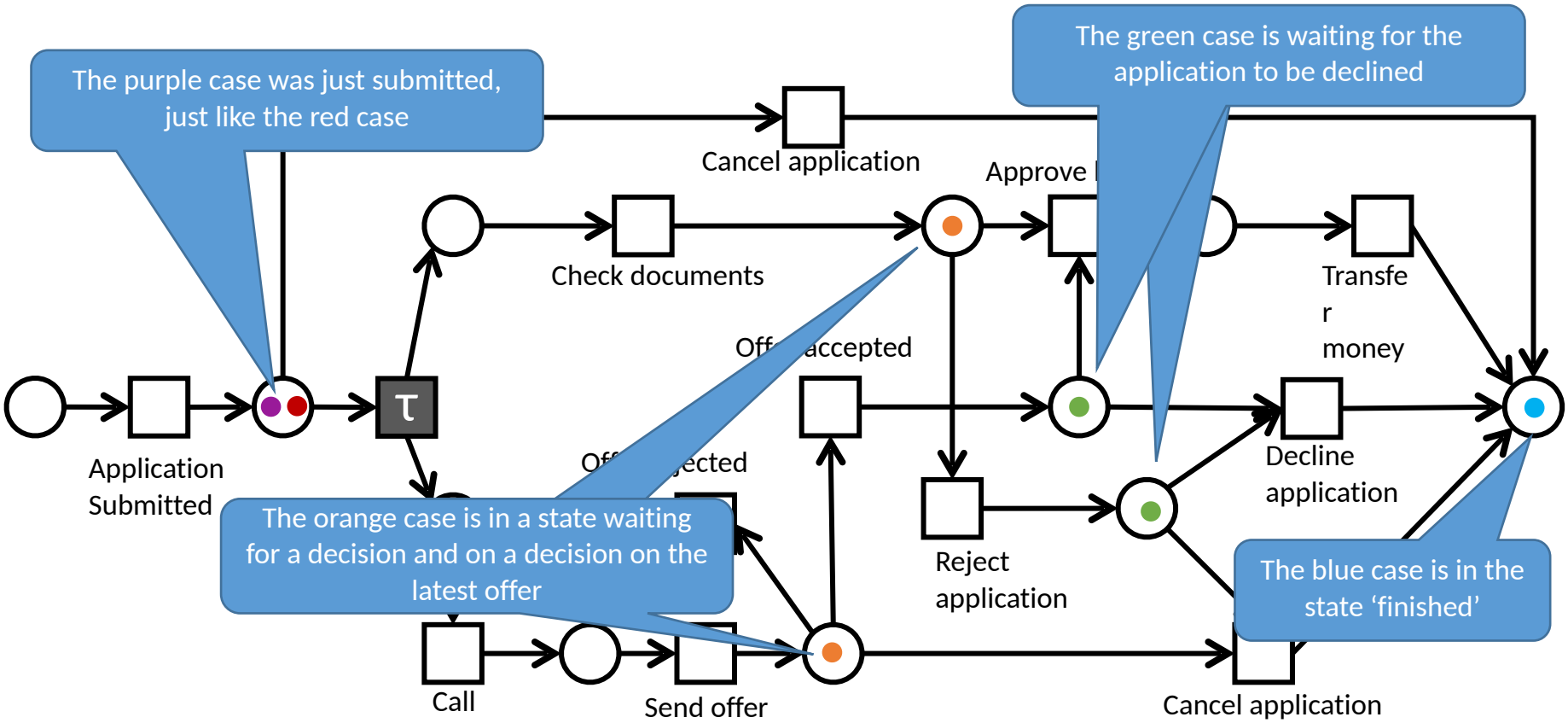


# System models vs. Process models

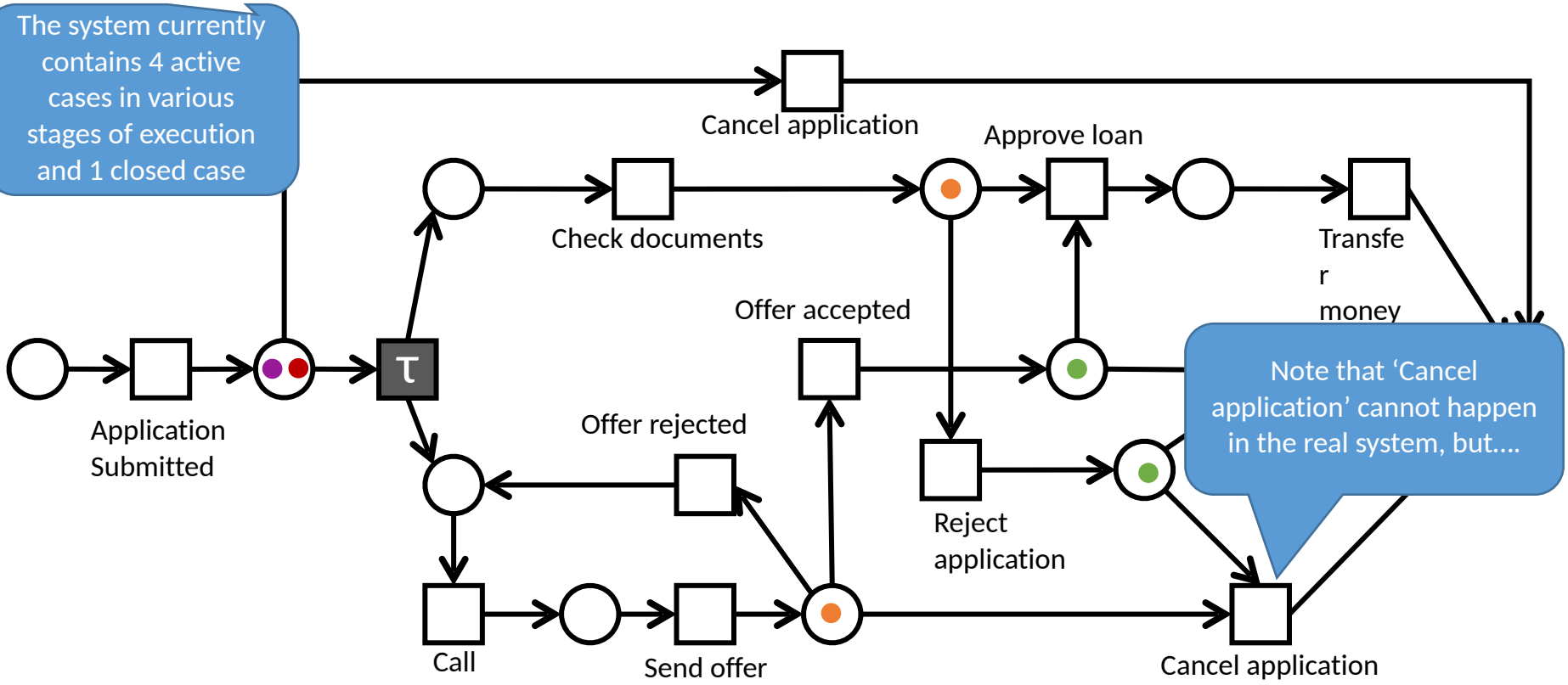
- When Petri nets are used to model systems, the marking denotes the state of the system.

- When Petri nets are used to model processes, the marking typically denotes the state of an individual case (and only markings are considered that can be reached from the initial marking with a single token in the initial place)

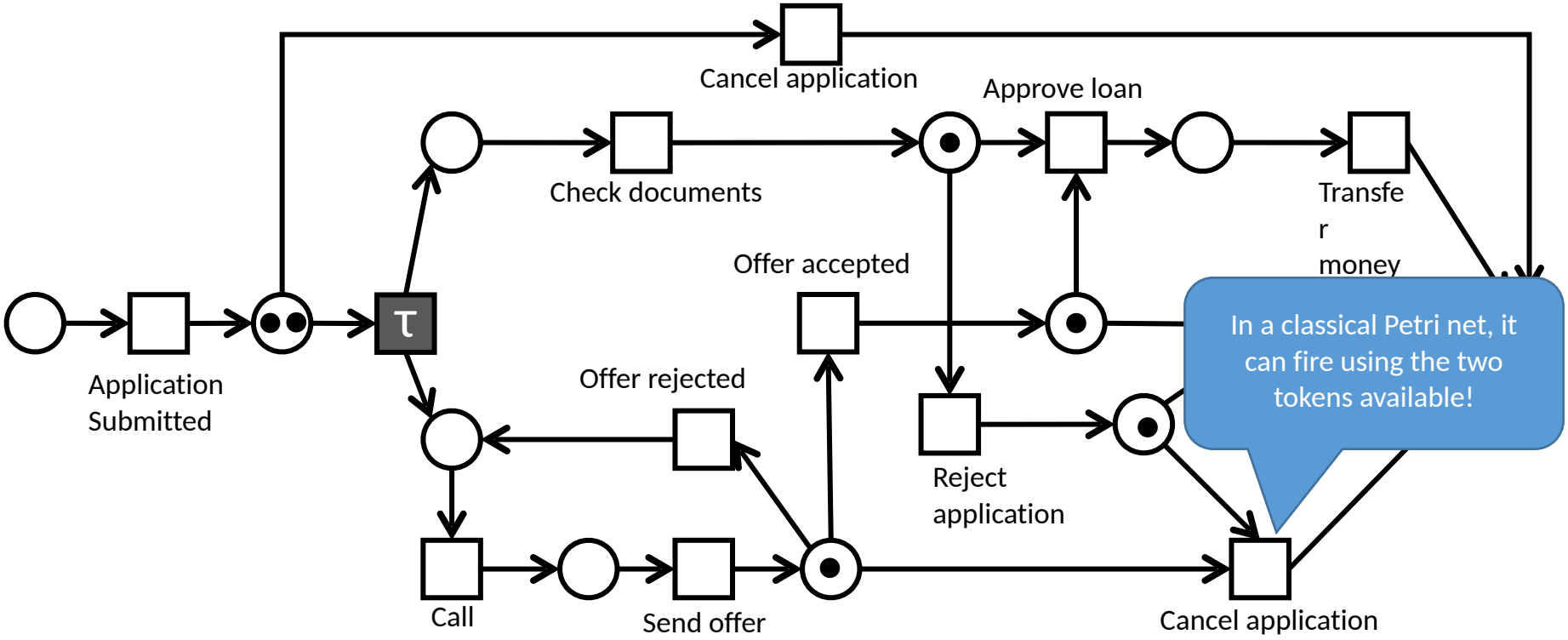
# State of a case vs. State of a system



# State of a case vs. State of a system



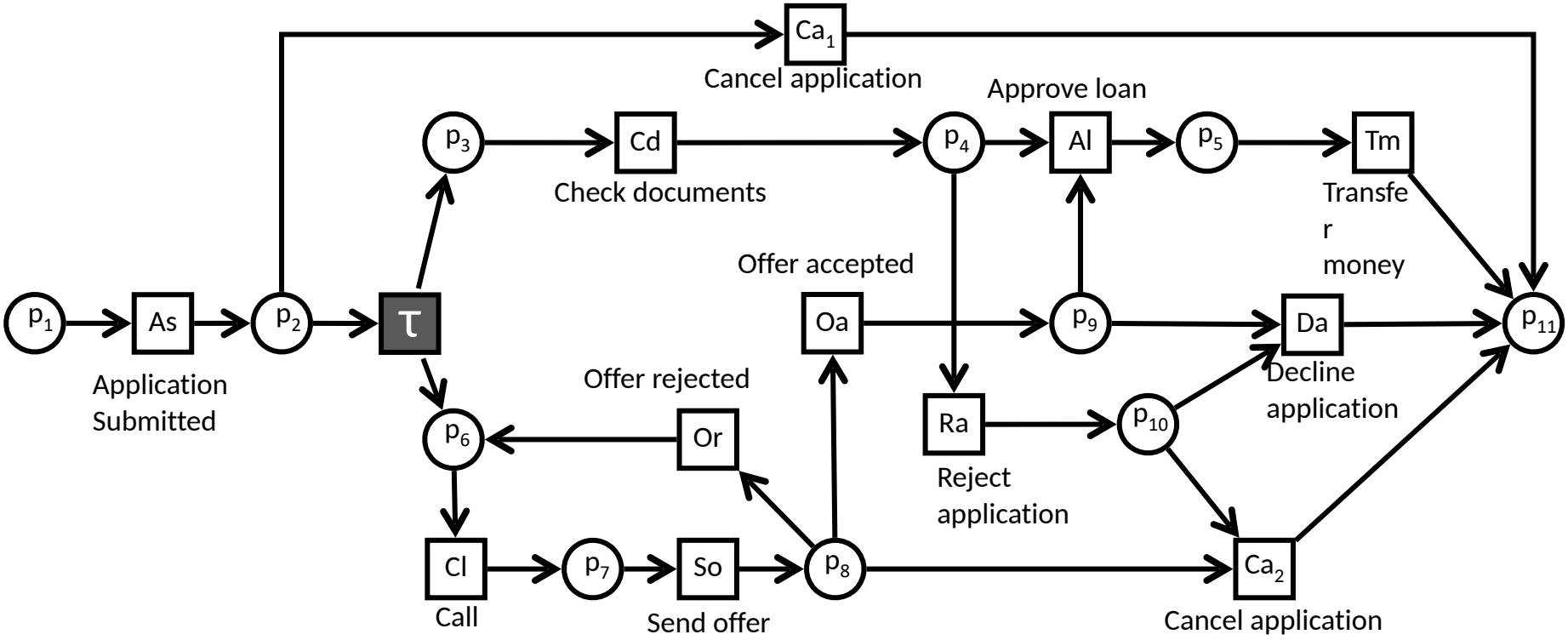
# State of a case vs. State of a system



# State Space of a Petri Net

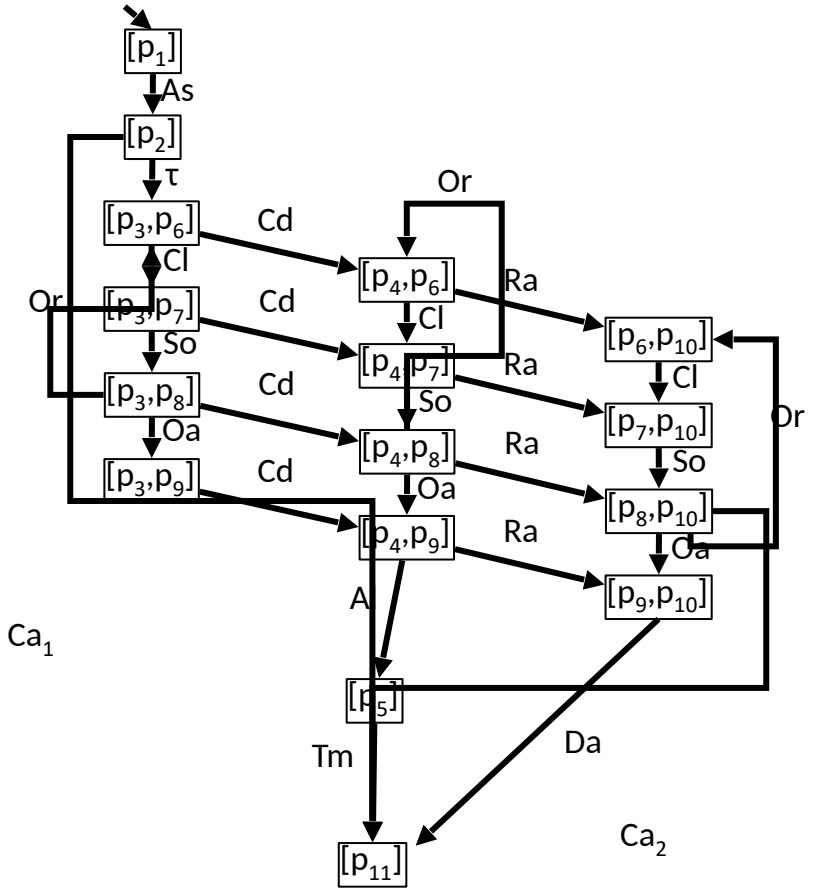
- Start from the initial marking
- Fire transitions and note down all new markings
- Do the same for all markings until all have been investigated

# State Space of a Petri Net: Label the sequences!



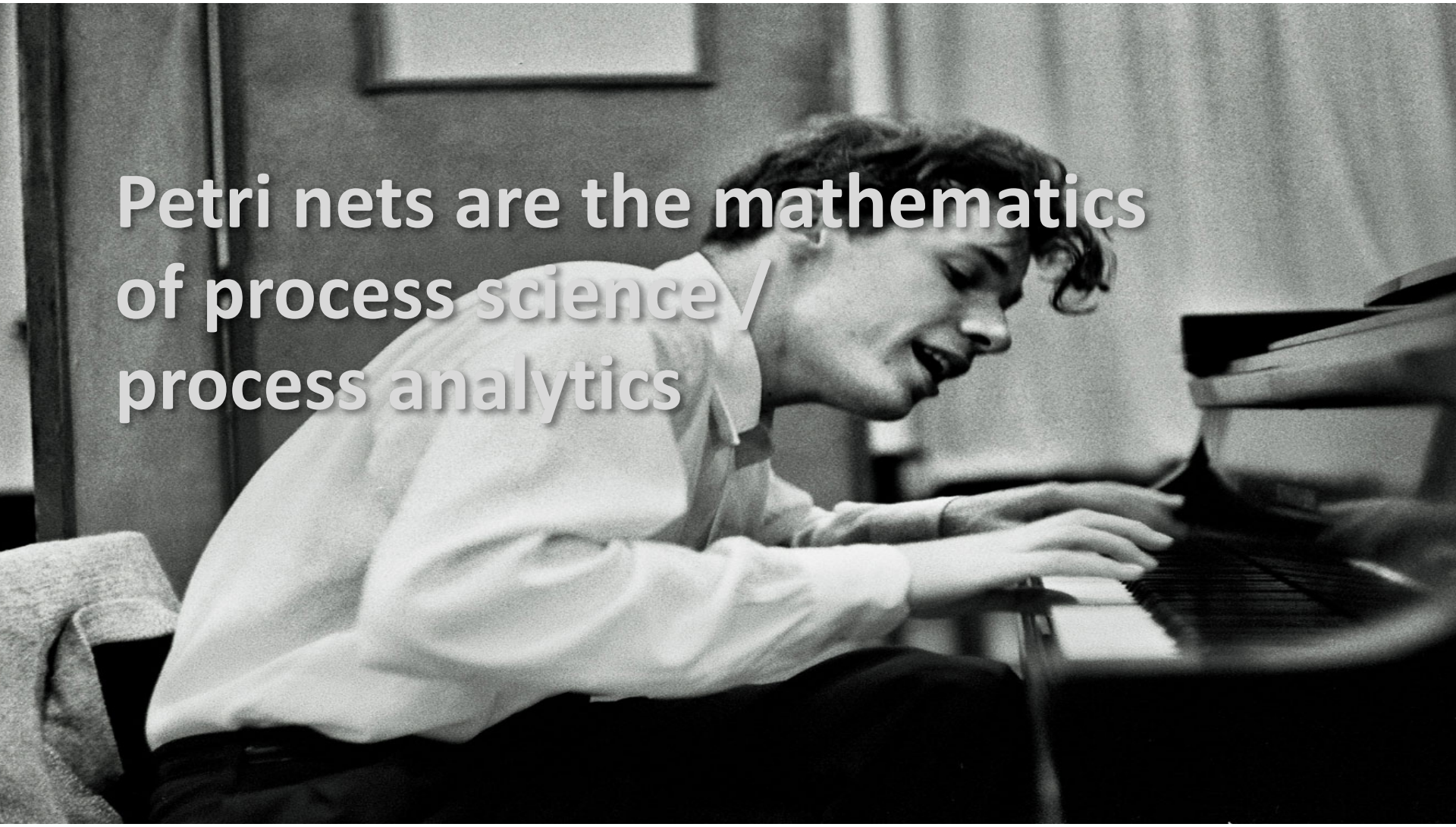
# State Space of a Petri Net

- Start from the initial marking
- Fire transitions and note down all new markings
- Do the same for all markings until all have been investigated




Statespace (transition system or marking graph) of the Petri net for simplified BPI12 example

Do you understand this process?



# Petri nets are the mathematics of process science / process analytics

# Take Away

Petri nets are a graphical notation supported by a sound mathematical ground

They naturally are able to describe systems and in particular, processes

Tool support at different levels