PV 2014

Complexity Bounds for Software Component Reconfiguration

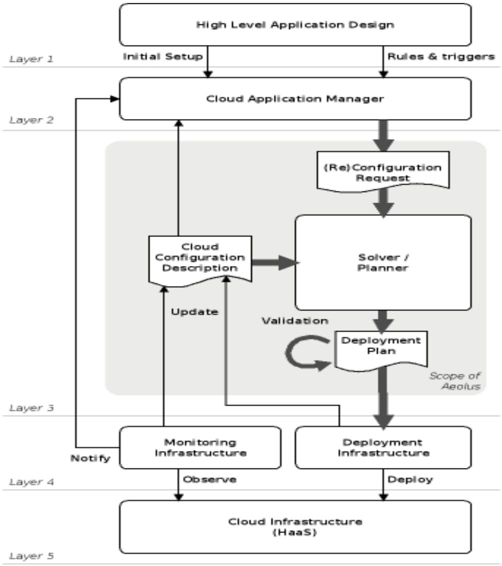
Jacopo Mauro & Gianluigi Zavattaro

Rome, Sept. 2014

Aeolus

Mastering the Complexity of the Cloud

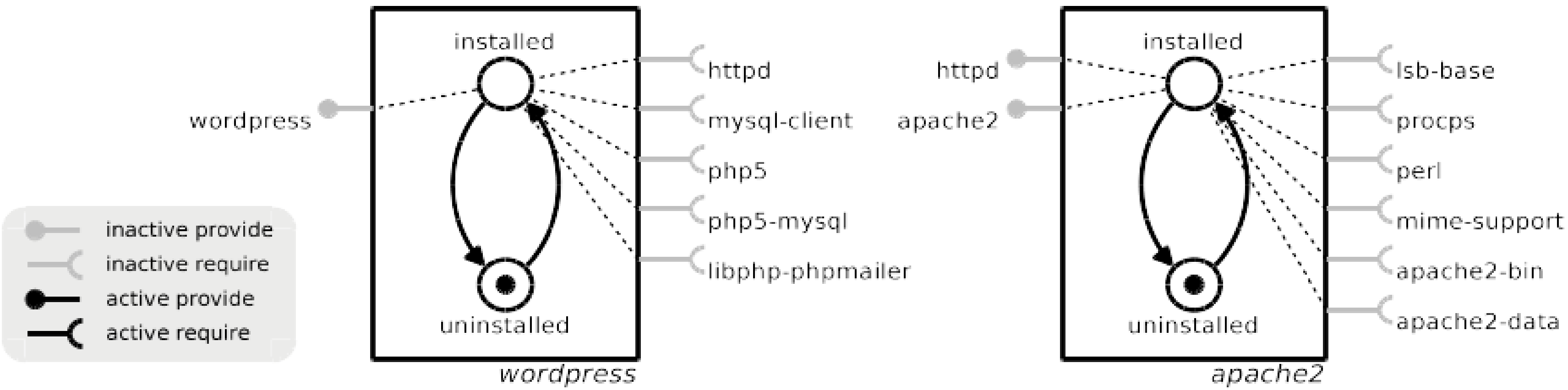
Models, languages and tools for the administrator of cloud applications



Structure of the talk

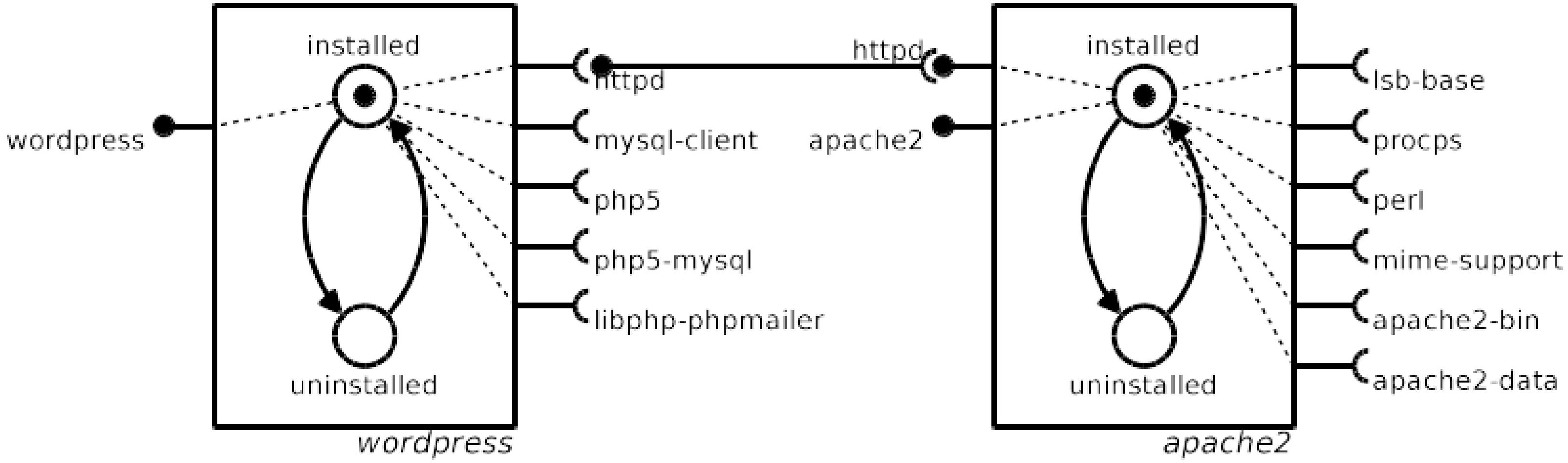
- **Brief summary of AEOLUS Model**
- What we knew so far about Deployment ?
- What about Reconfiguration ?
- Conclusions

Aeolus components



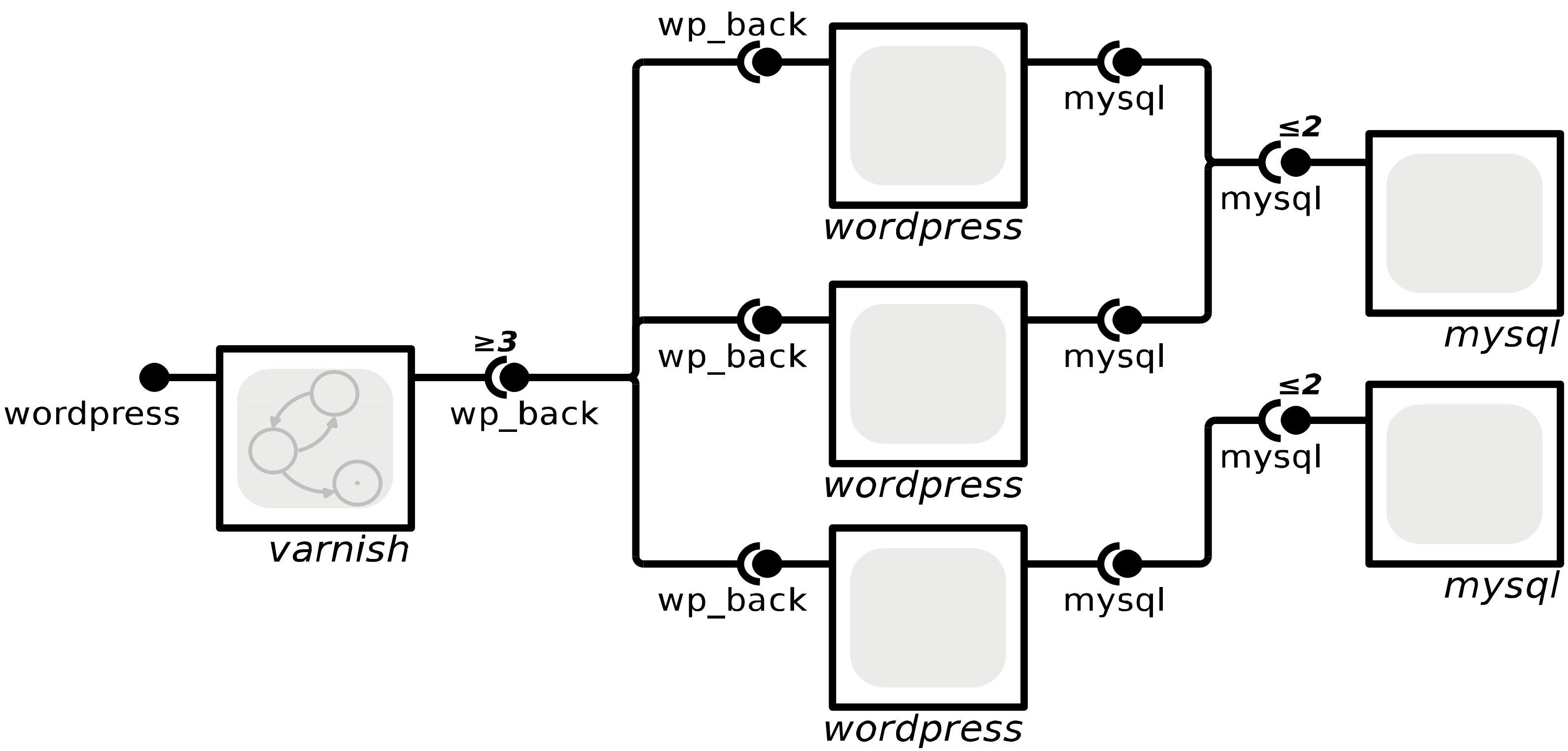
Bindings

Binding between two packages



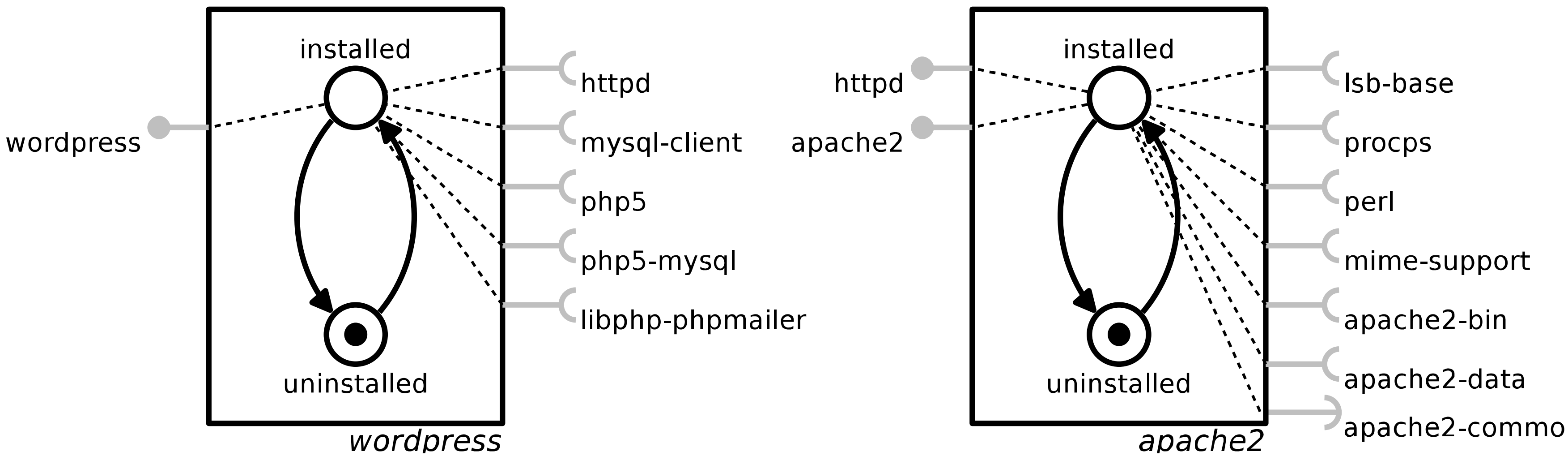
Capacity Planning

Capacity constraints: upper and lower bounds to provide and require ports (resp.)



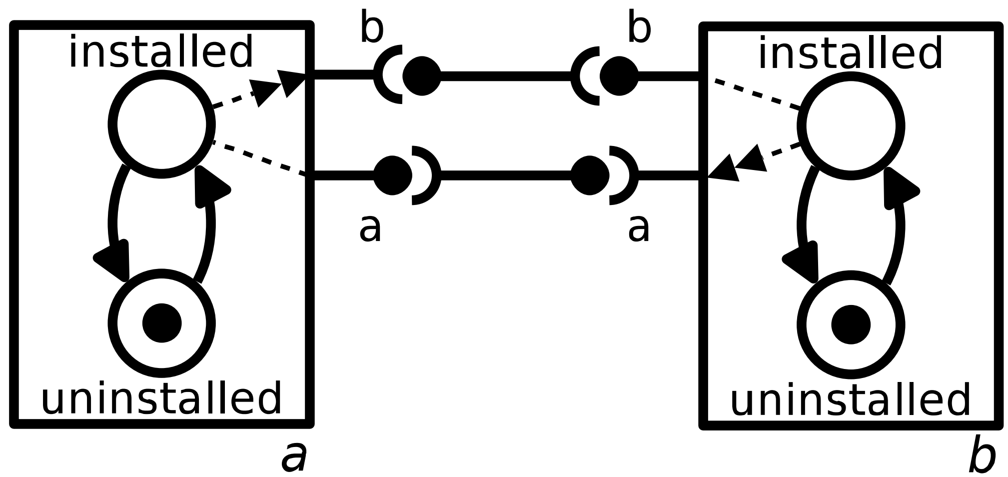
Conflicts

Conflict (no other components can activate that port)



Multi state changes

In some specific case, actions must be executed simultaneously



Actions

- Create
- Delete
- State Change
- Bind
- Unbind

Deployment problem

- **Input**:
  - A set of components to use
  - Target: component + state
- **Output**:
  - ***yes*** exists a plan from **empty configuration** to a configuration containing a **target**
  - ***no*** otherwise

Reconfiguration problem

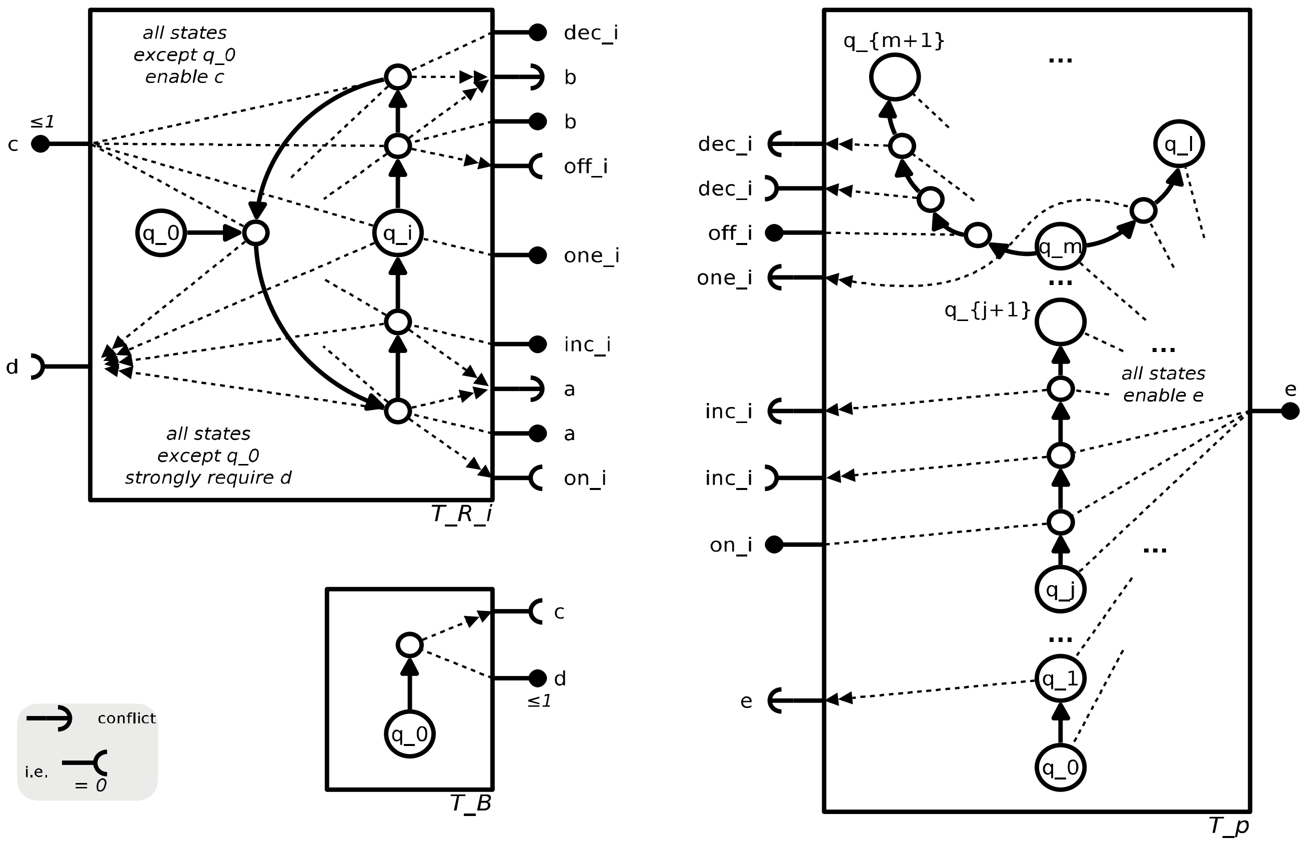
Reconfiguration problem: like deployment problem but with a given **initial configuration**

Structure of the talk

- Brief summary of AEOLUS Model
- **What we knew so far about Deployment ?**
- What about Reconfiguration ?
- Conclusions

Undecidability of Aeulus

The deployment problem is **undecidable** in Aeolus



Aeolus-

Deployment problem **Polytime** without capacity constraints, conflicts, multi state change.

Algorithm 1 Checking achievability in the Aeolus$^-$ model

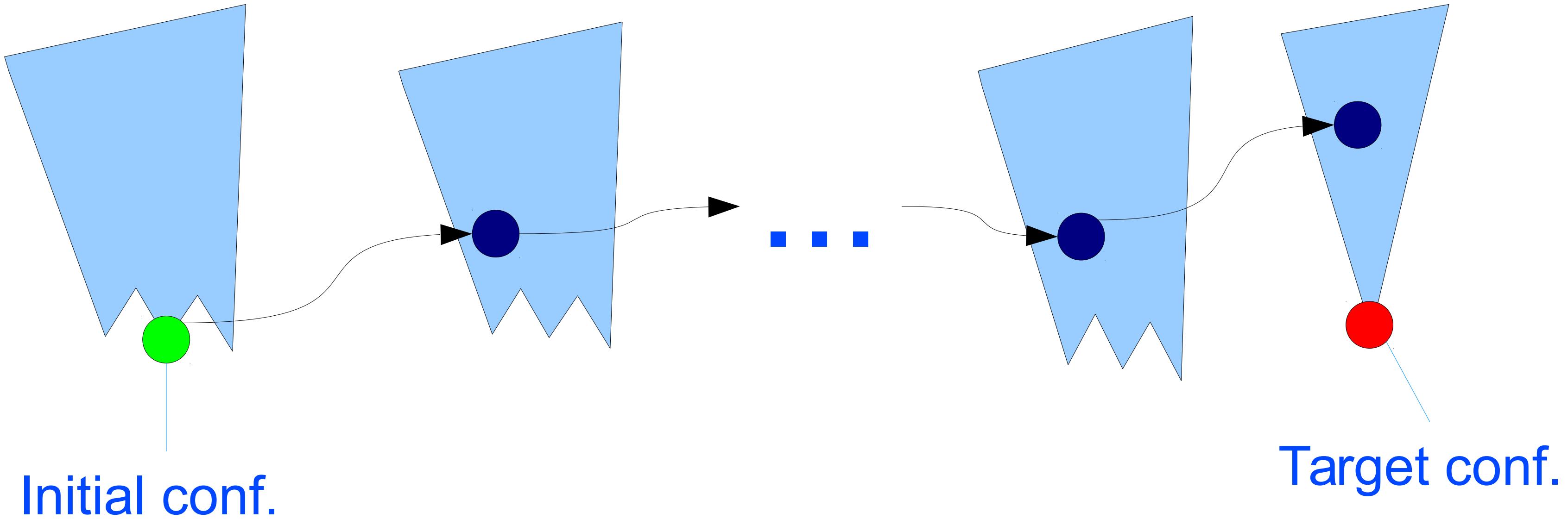
function ACHIEVABILITY(universe of resources U, resource type $\mathscr{T}$, state q)
  $absConf := \{\langle \mathscr{T}', \mathscr{T}'.\texttt{init}\rangle \mid \mathscr{T}' \in U\}$
  $provPort := \bigcup_{\langle \mathscr{T}', q'\rangle \in absConf} \{dom(\mathscr{T}'.\mathbf{P}\texttt{map}(q'))\}$
  repeat
    $new := \{\langle \mathscr{T}', q'\rangle \mid \langle \mathscr{T}', q''\rangle \in absConf, (q'', q') \in \mathscr{T}'.\texttt{trans}\} \setminus absConf$
    $newPort := \bigoplus_{\langle \mathscr{T}', q'\rangle \in new} \{\!\{dom(\mathscr{T}'.\mathbf{P}\texttt{map}(q'))\}\!\}$
    while $\exists \langle \mathscr{T}', q'\rangle \in new$ s.t. $dom(\mathscr{T}'.\mathbf{R}_s\texttt{map}(q')) \not\subseteq provPort \cup newPort$ **do**
      $new := new \setminus \{\langle \mathscr{T}', q'\rangle\}$
      $newPort := newPort \ominus \{\!\{dom(\mathscr{T}'.\mathbf{P}\texttt{map}(q'))\}\!\}$
    end while
    $absConf := absConf \cup new$
    $provPort := provPort \cup newPort$
  until $new = \emptyset$
  if $\langle \mathscr{T}, q\rangle \in absConf$ and $dom(\mathscr{T}.\mathbf{R}_w\texttt{map}(q)) \subseteq provPort$ **then return true**
  else return false
  end if
end function

Structure of the talk

- Brief summary of AEOLUS Model
- What we knew so far about Deployment ?
- **What about Reconfiguration ?**
- Conclusions

Aeolus Core

- Conflicts but no capacity constraints (~ packages)
- Reconfiguration problem → **Decidable**
- Proof → **WSTS**

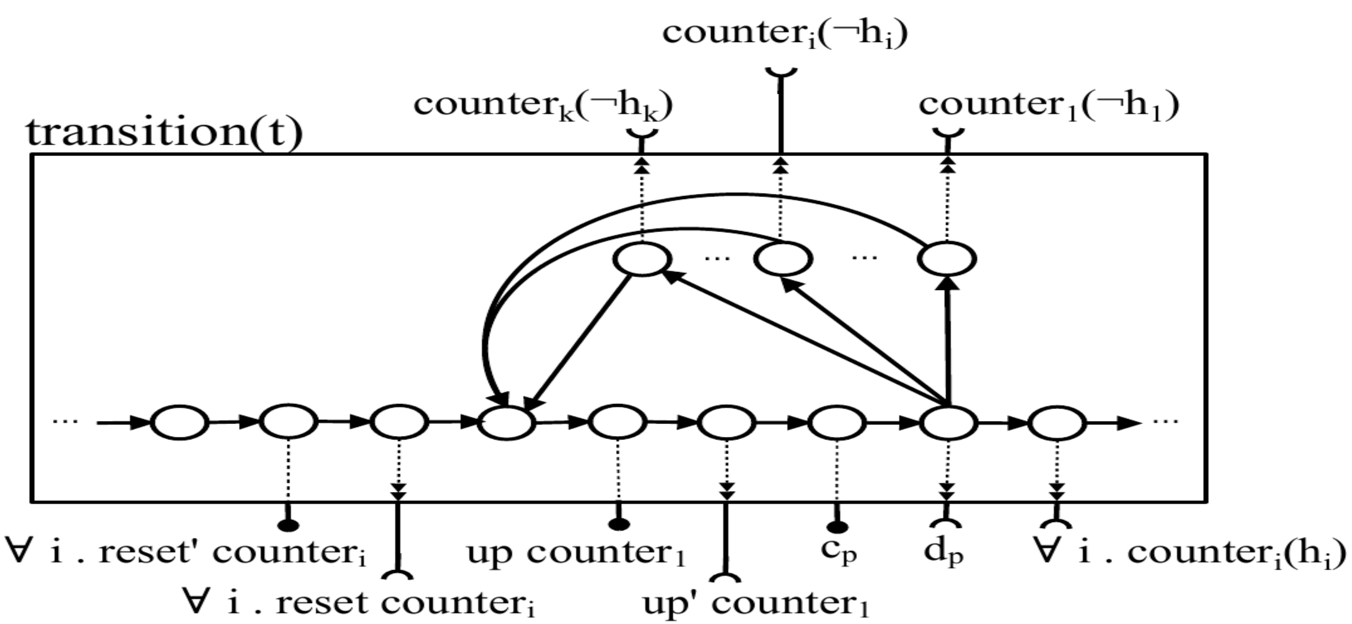


The WQO

- C1 $\leqslant$ C2:
  - No component r in state s in C1 $\rightarrow$ No component r in state s in C2
  - 1 component r in state s in C1 $\rightarrow$ 1 component r in state s in C2
  - x components r in state s in C1 $\rightarrow$ y > x components r in state s in C2

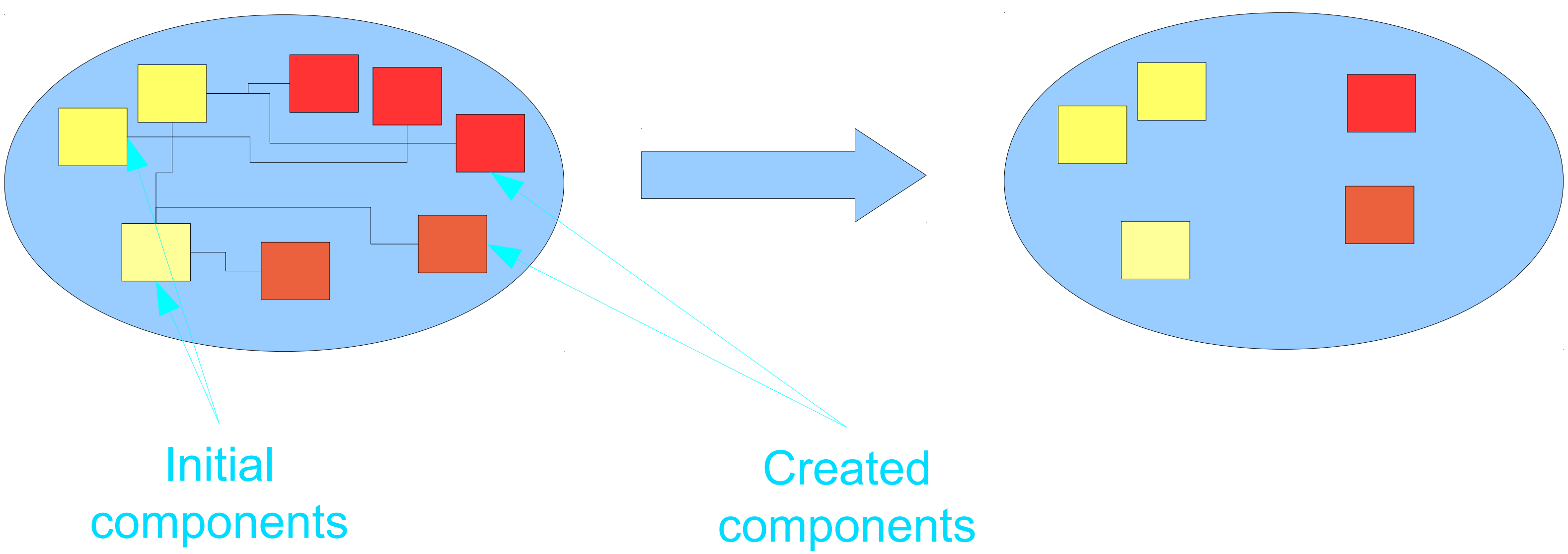
Aeolus Core

- Reconfiguration problem → **Ackerman hard**
- Proof → encoding the **Coverability** of Petri Nets with Reset Arcs



Aeolus-

- No capacity, conflicts, multi state change
- Reconfiguration problem → **PSpace**
- Proof → abstract from bindings & newly created components



Aeolus-

- Compute possible abstract plan using **non deterministic Turing machine**
- Space used $\rightarrow$ polynomial

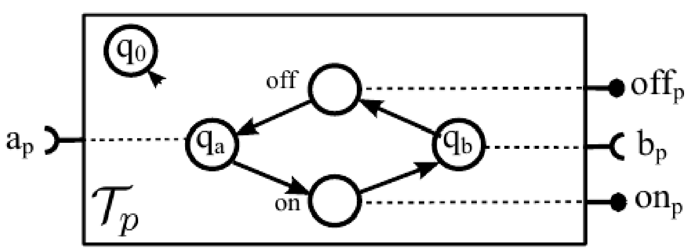
Algorithm 1 Check reconfiguration for $\mathcal{C}_0 = \langle U, Z_0, S, B\rangle$ and target $\mathcal{T}_t, q_t$

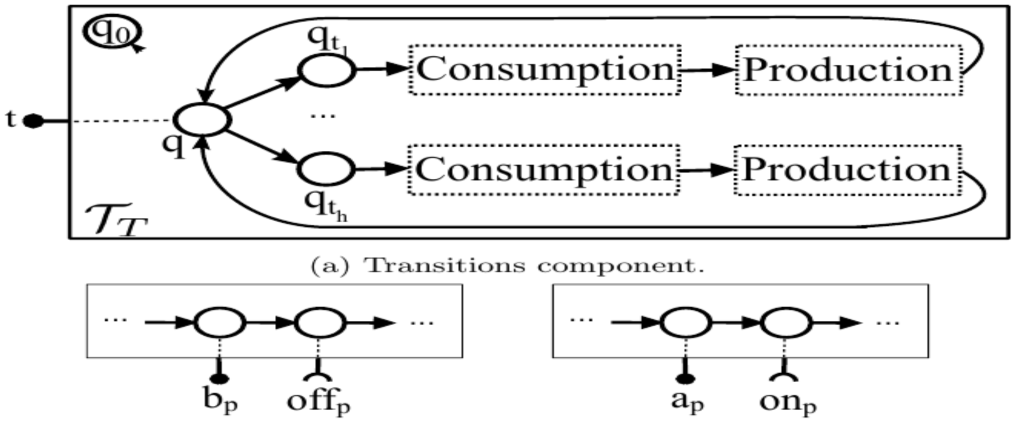
---

for all $\langle \mathcal{T}, q\rangle$ pairs in the universe U **do**
  $\mathcal{B}_i(\langle \mathcal{T}, q\rangle) = \mathcal{C}^{\#}_{\langle \mathcal{T}, q\rangle}(Z_0)$
  $\mathcal{B}_c(\langle \mathcal{T}, q\rangle) = \mathit{False}$
$\mathit{counter} = 0$
while $\mathit{counter} \leq |Z_0|^k * 2^k$ **do** $\triangleright$ k is the number of $\langle \mathcal{T}, q\rangle$ pairs in U
  guess $\mathcal{B}'_i, \mathcal{B}'_c$
  if $\langle \mathcal{B}_i, \mathcal{B}_c\rangle \not\rightarrow \langle \mathcal{B}'_i, \mathcal{B}'_c\rangle$ **then return** False
  if $\mathcal{B}'_i(\mathcal{T}_t, q_t) > 0$ *or* $\mathcal{B}'_c(\mathcal{T}_t, q_t)$ **then return** True
  $\mathit{counter} = \mathit{counter} + 1$; $\mathcal{B}_i = \mathcal{B}'_i$; $\mathcal{B}_c = \mathcal{B}'_c$
return False

Aeolus-

- Reconfiguration → **PSpace hard**
- Proof → encoding the **reachability** problem of 1-Safe Petri Nets





(a) Transitions component.

(b) Consumption phase of 1 token from place p

(c) Production phase of 1 token from place p

Structure of the talk

- Brief summary of AEOLUS Model
- What we knew so far about Deployment ?
- What about Reconfiguration ?
- **Conclusions**

Conclusions

- Deployment problem in Aeolus- polynomial $\longrightarrow$ **METIS**
- Extensions of METIS for reconfiguration $\longrightarrow$ yes but possibly lose completeness
- Hint to deal with reconfiguration: abstract plan with ad-hoc **heuristics**
- Other “usable” Aeolus model fragments