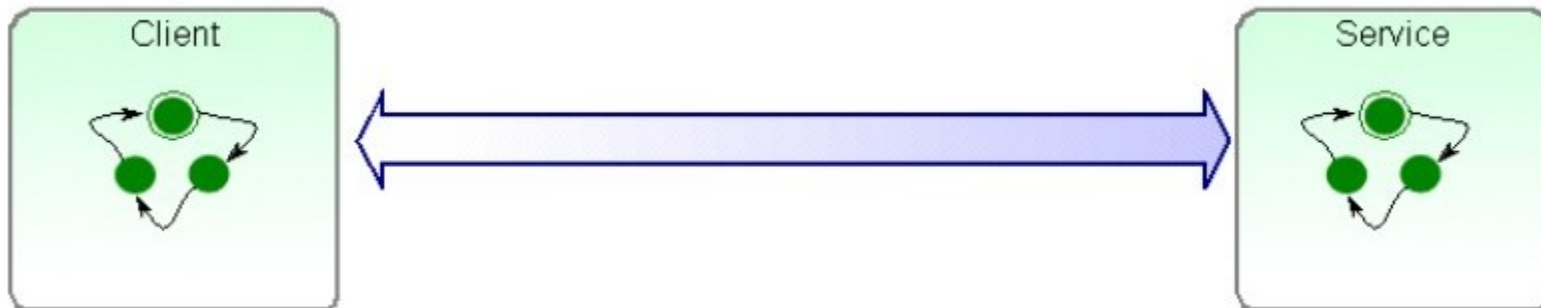


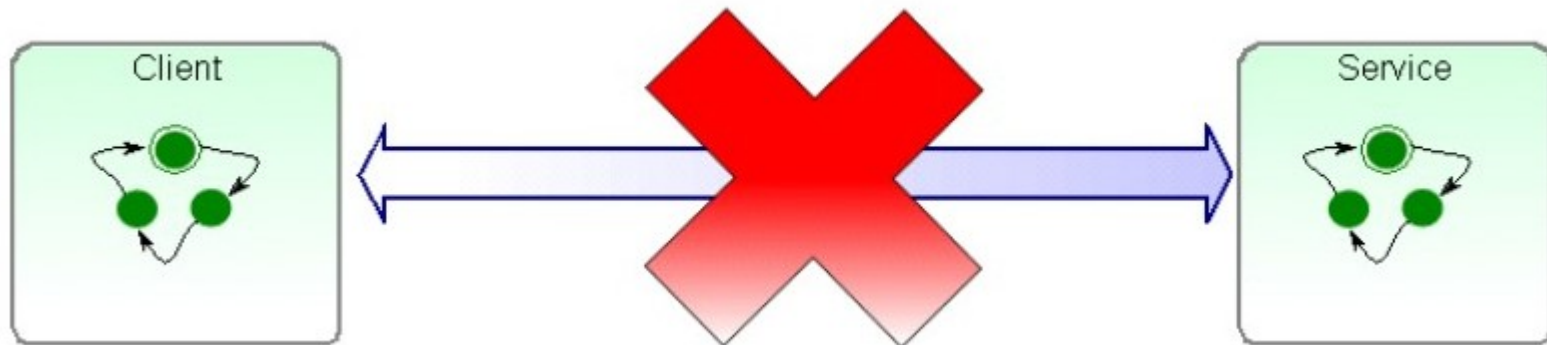


# Semi-Automatic Specification of Behavioural Service Adaptation Contracts

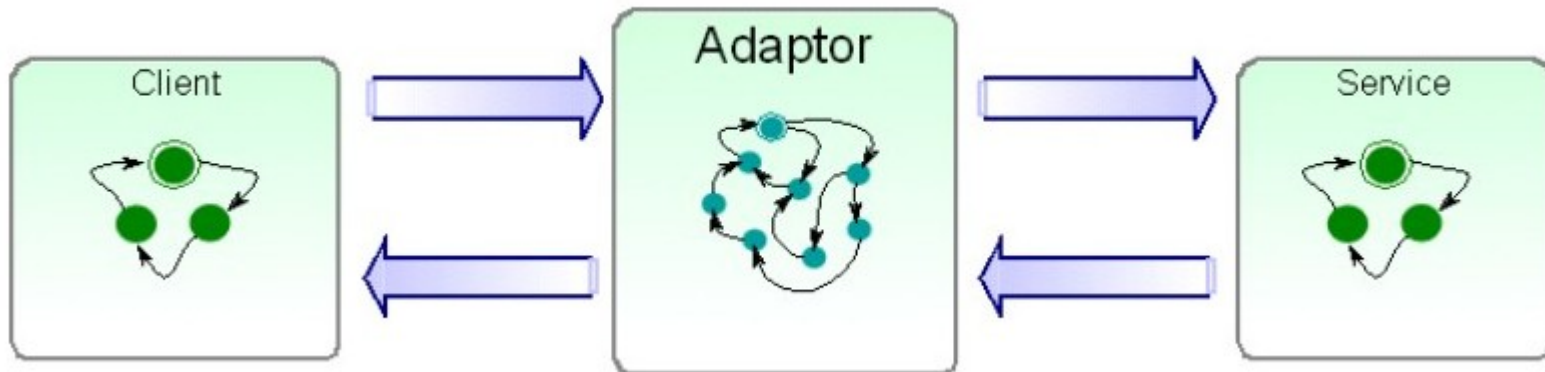
J. Cámara, **J. Antonio Martín**, G. Salaün,  
C. Canal, E. Pimentel

FESCA'10  
27 March  
Paphos, Cyprus

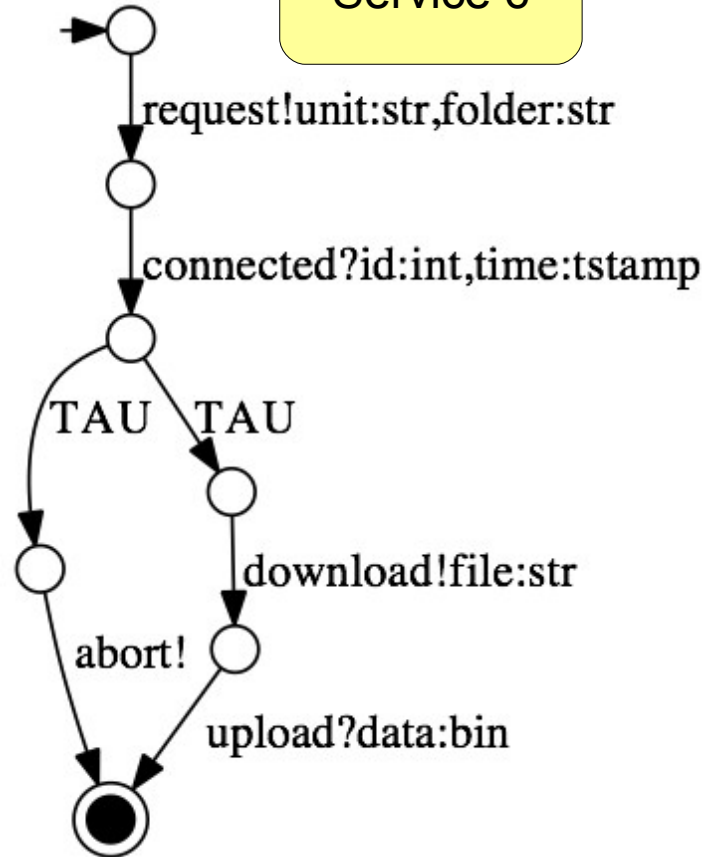




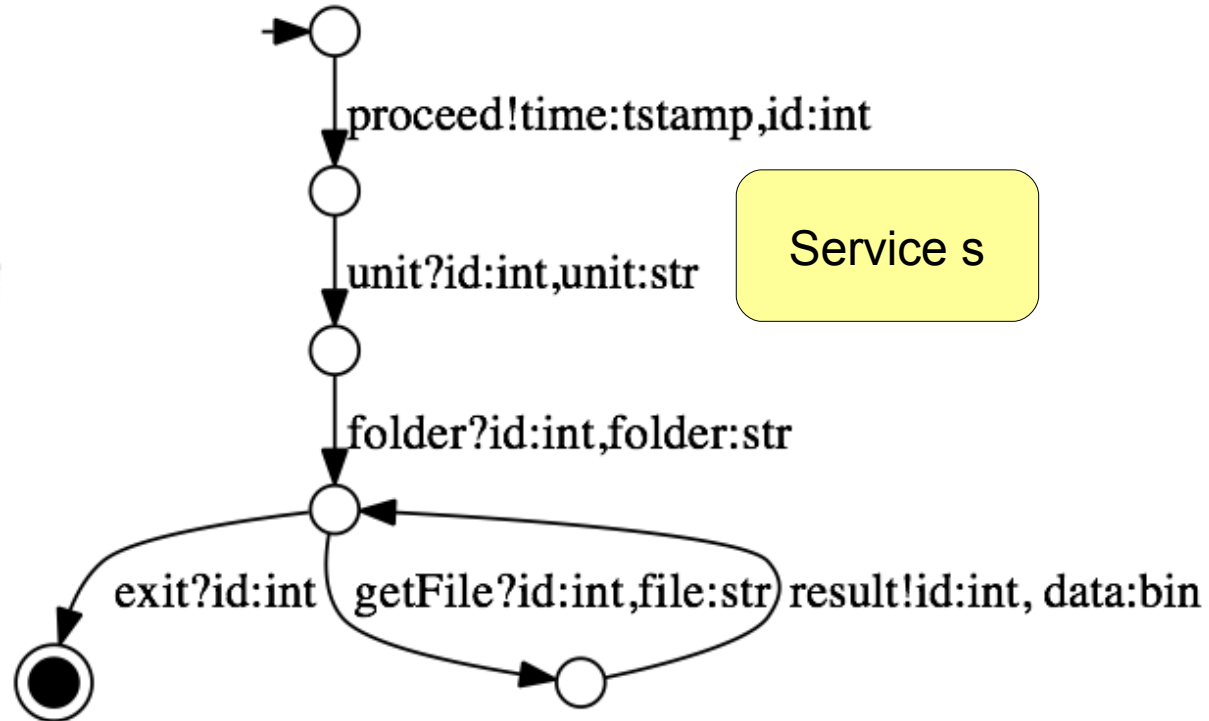
# Service Adaptation



Service c



Service s

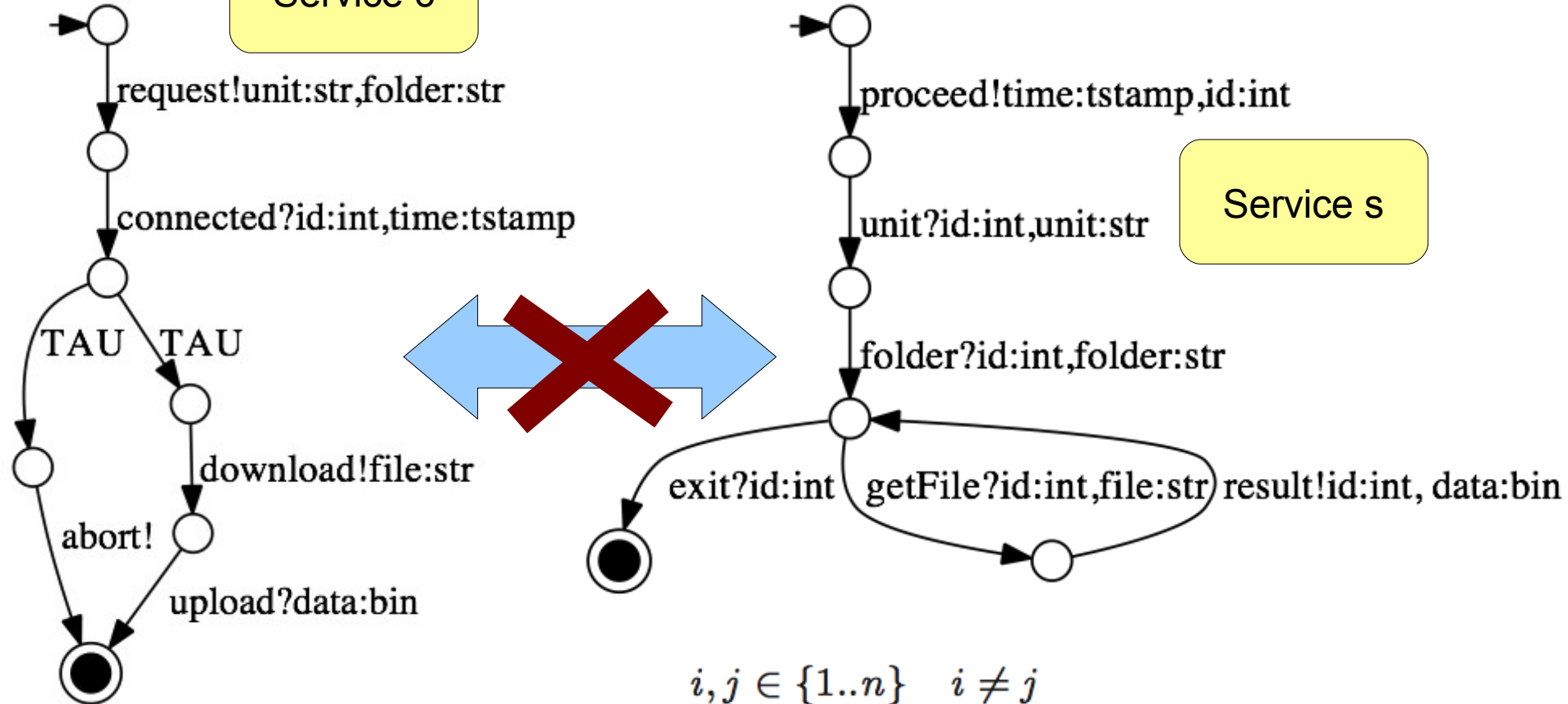


**Definition** [STS] A Symbolic Transition System or STS is a tuple  $(A, S, I, F, T)$  where:  $A$  is an alphabet which corresponds to the set of labels associated to transitions,  $S$  is a set of states,  $I \in S$  is the initial state,  $F \subseteq S$  are the final states, and  $T : S \times A \times S$  is the transition function.

# Incompatible Services

Service c

Service s

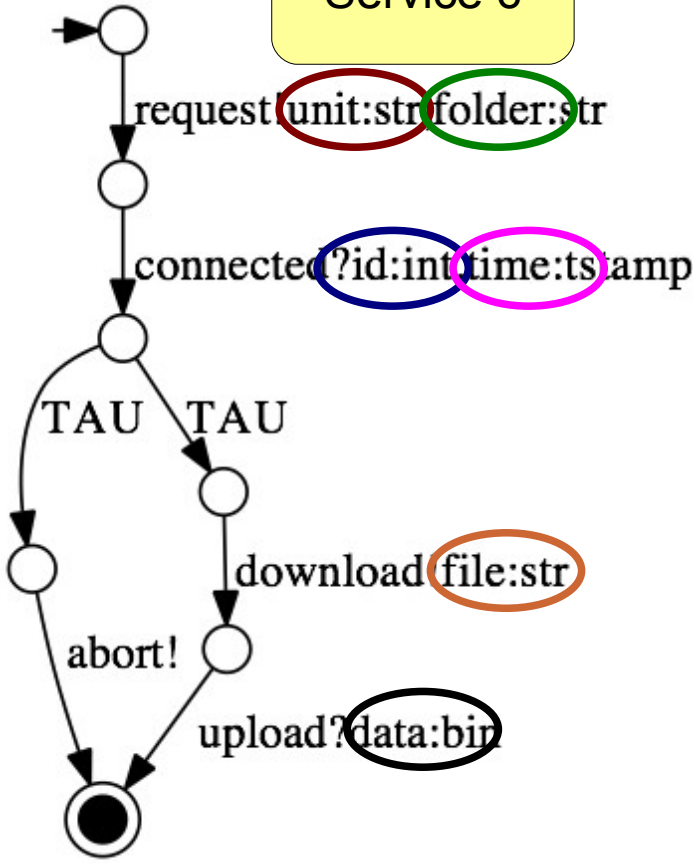


$$\begin{aligned}
 & i, j \in \{1..n\} \quad i \neq j \\
 & \langle s_i, E_i \rangle \xrightarrow{a!v}_b \langle s'_i, E_i \rangle \quad \langle s_j, E_j \rangle \xrightarrow{a?x}_b \langle s'_j, E_j \rangle \\
 & \text{type}(x) = \text{type}(v) \quad E'_j = E_j \circ \langle x, v \rangle
 \end{aligned}$$

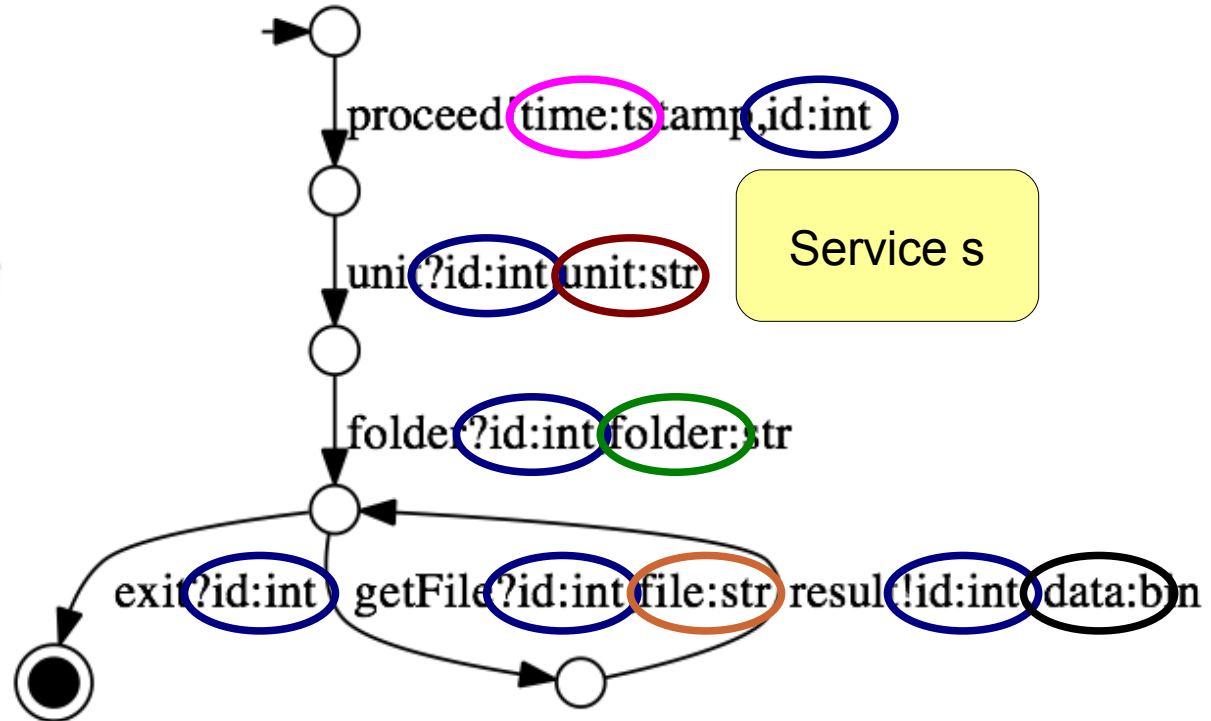
$$\{as_1, \dots, \langle s_i, E_i \rangle, \dots, \langle s_j, E_j \rangle, \dots, as_n\} \xrightarrow{a!v}_c \{as_1, \dots, \langle s'_i, E_i \rangle, \dots, \langle s'_j, E'_j \rangle, \dots, as_n\}$$

# Behavioural Incompatibilities

Service c



Service s

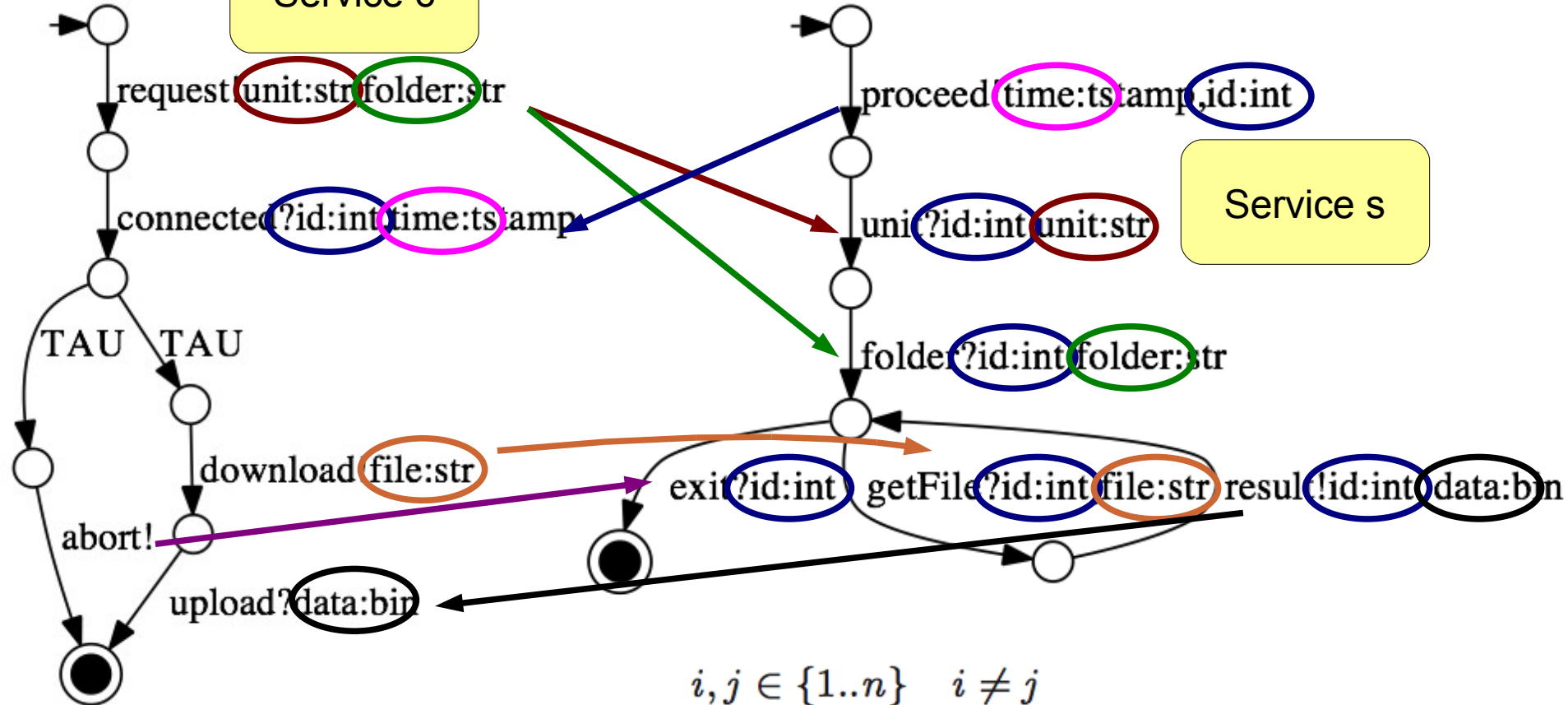


$$\begin{aligned}
 & i, j \in \{1..n\} \quad i \neq j \\
 & \langle s_i, E_i \rangle \xrightarrow{a!v}_b \langle s'_i, E_i \rangle \quad \langle s_j, E_j \rangle \xrightarrow{a?x}_b \langle s'_j, E_j \rangle \\
 & \text{type}(x) = \text{type}(v) \quad E'_j = E_j \circ \langle x, v \rangle
 \end{aligned}$$

---


$$\{as_1, \dots, \langle s_i, E_i \rangle, \dots, \langle s_j, E_j \rangle, \dots, as_n\} \xrightarrow{a!v}_c \{as_1, \dots, \langle s'_i, E_i \rangle, \dots, \langle s'_j, E'_j \rangle, \dots, as_n\}$$

# Behavioural Incompatibilities

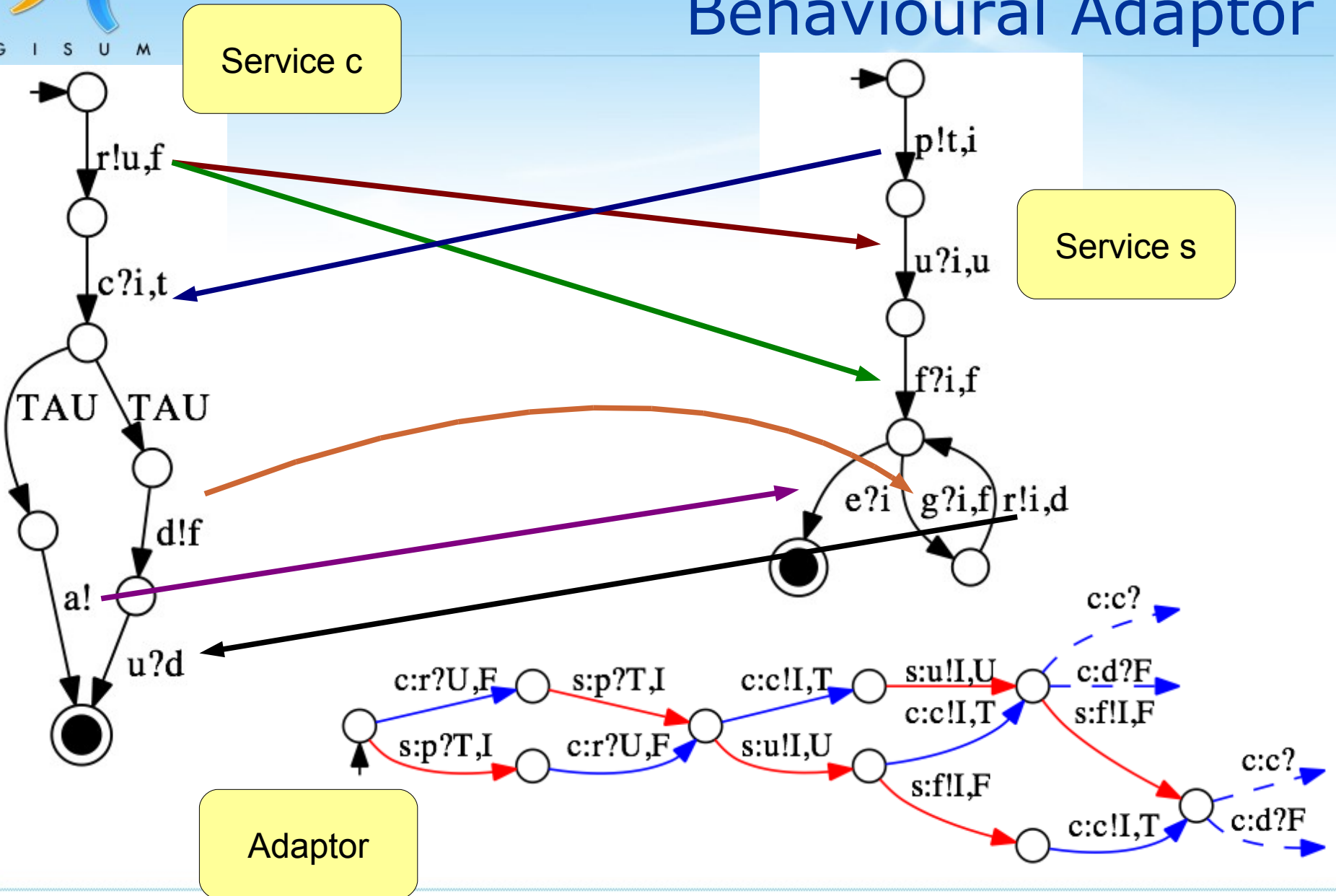


$$\begin{aligned}
 & i, j \in \{1..n\} \quad i \neq j \\
 & \langle s_i, E_i \rangle \xrightarrow{a!v}_b \langle s'_i, E_i \rangle \quad \langle s_j, E_j \rangle \xrightarrow{a?x}_b \langle s'_j, E_j \rangle \\
 & \text{type}(x) = \text{type}(v) \quad E'_j = E_j \circ \langle x, v \rangle
 \end{aligned}$$

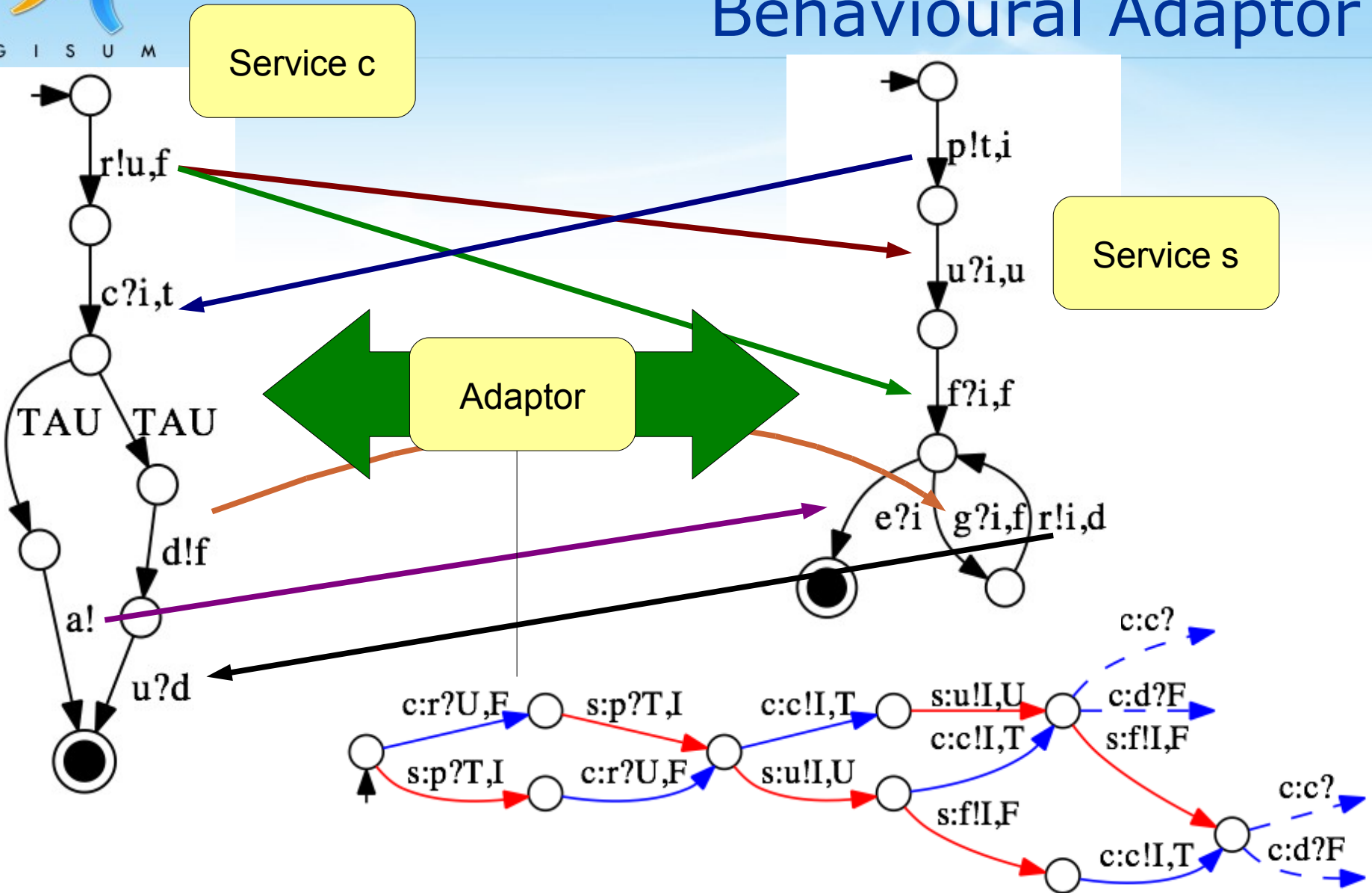
$$\{ as_1, \dots, \langle s_i, E_i \rangle, \dots, \langle s_j, E_j \rangle, \dots, as_n \} \xrightarrow{a!v}_c \{ as_1, \dots, \langle s'_i, E_i \rangle, \dots, \langle s'_j, E'_j \rangle, \dots, as_n \}$$



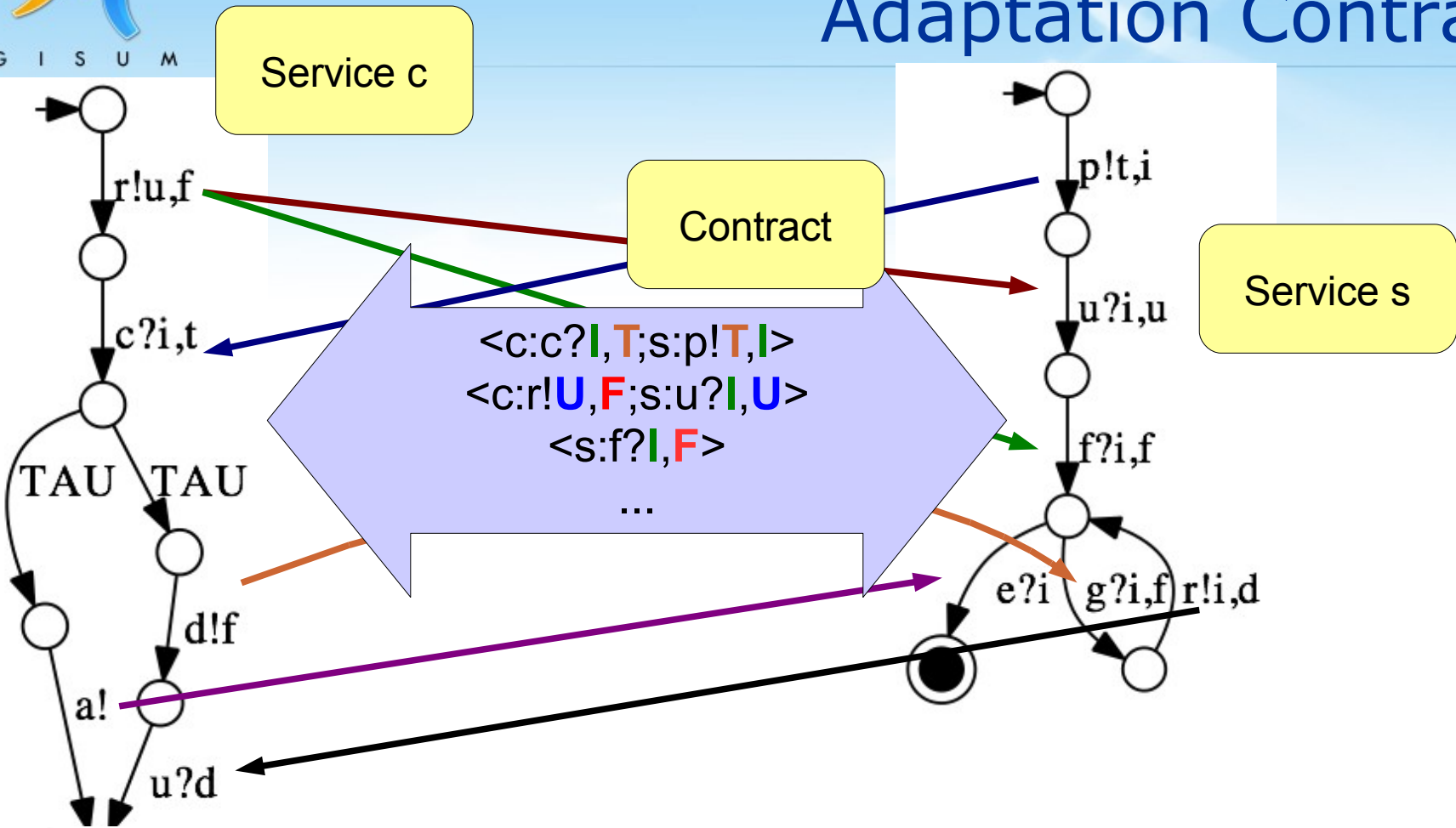
# Behavioural Adaptor



# Behavioural Adaptor

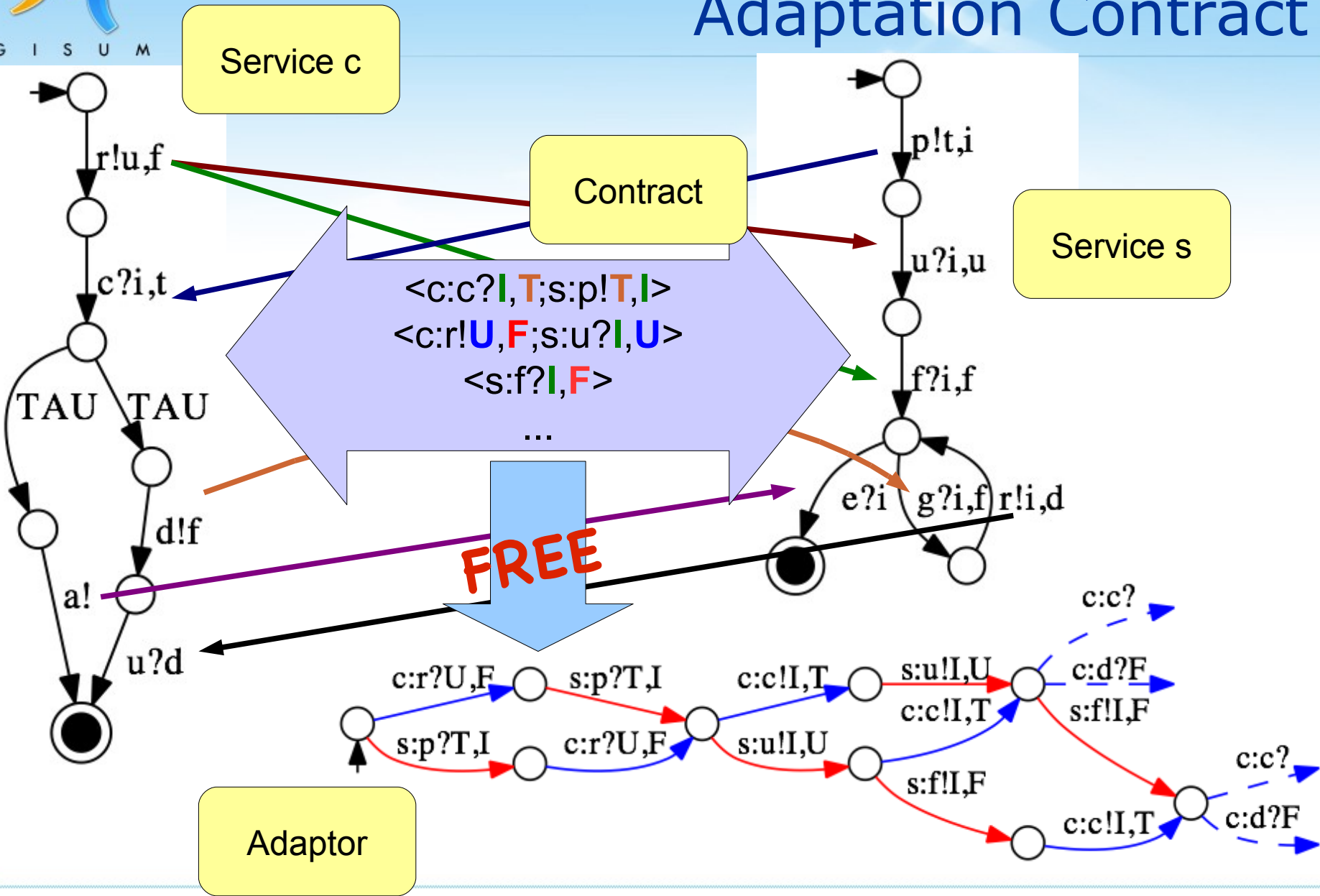


# Adaptation Contract



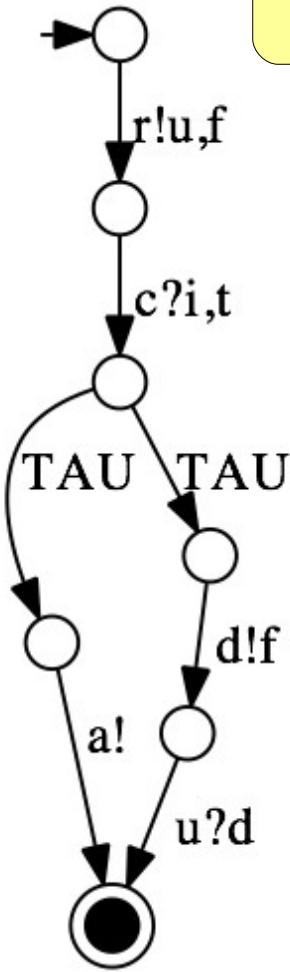
**Definition** [Vector] A vector for a set of service  $STS_i = (A_i, S_i, I_i, F_i, T_i), i \in \{1, \dots, n\}$  is an element of  $A_j \cup (A_j \times A_k)$  with  $j, k \in \{1, \dots, n\}, j \neq k$ . Such a vector is noted  $\langle s_j:l \rangle$ , or  $\langle s_j:l, s_k:l' \rangle$  where  $s_j, s_k$  are service identifiers, and  $l, l'$  are labels on the alphabets of services  $A_j, A_k$ , where message parameters are substituted by placeholders relating the arguments.

# Adaptation Contract

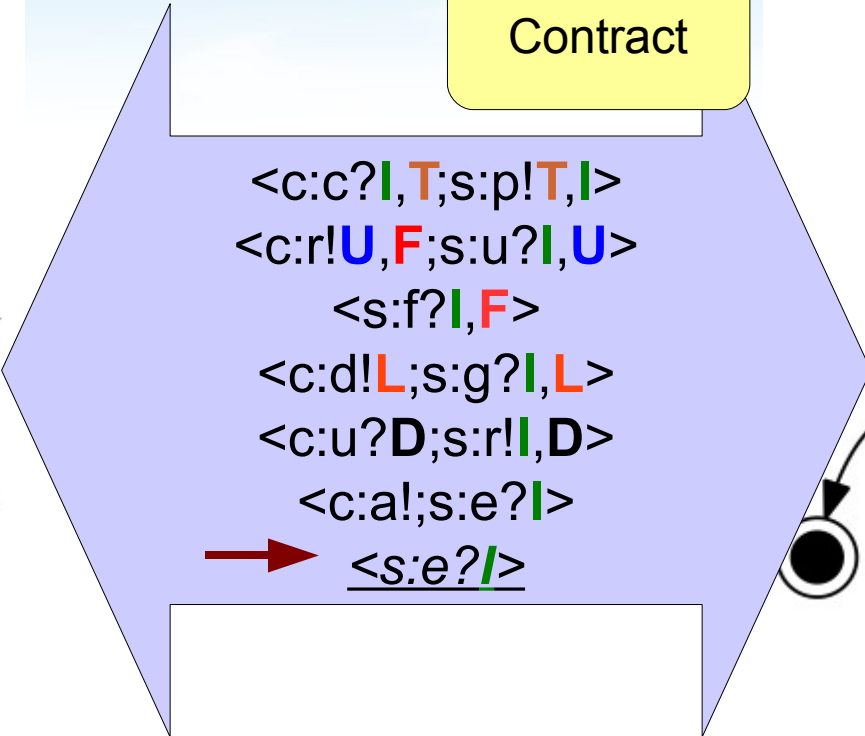


# Adaptation Contract

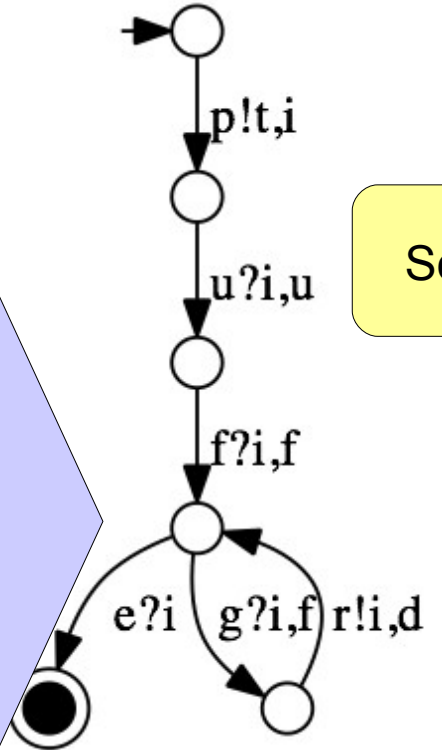
Service c



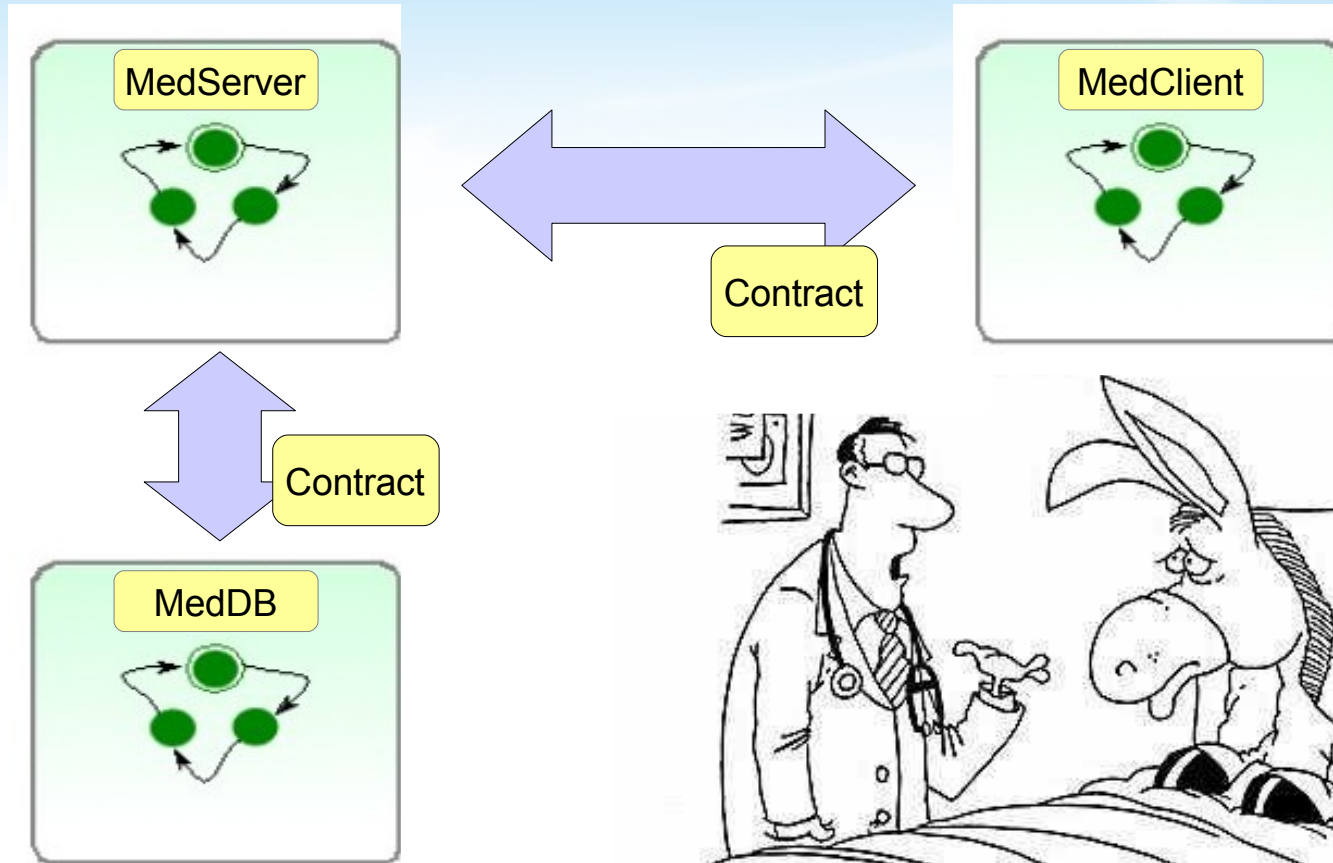
Contract



Service s

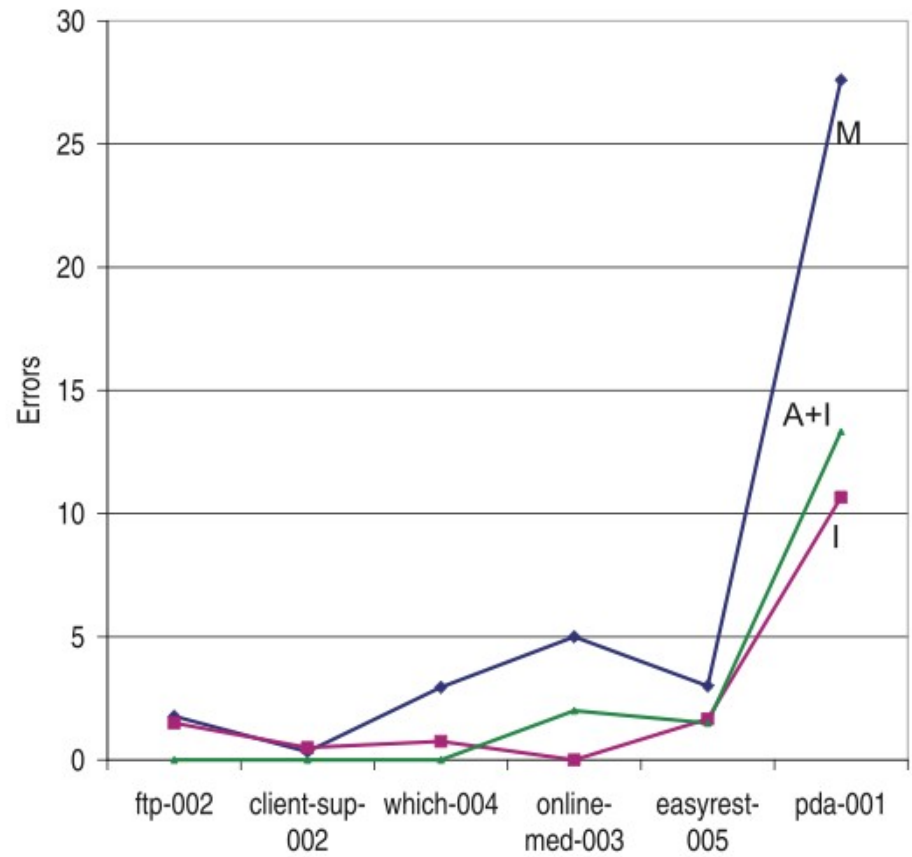
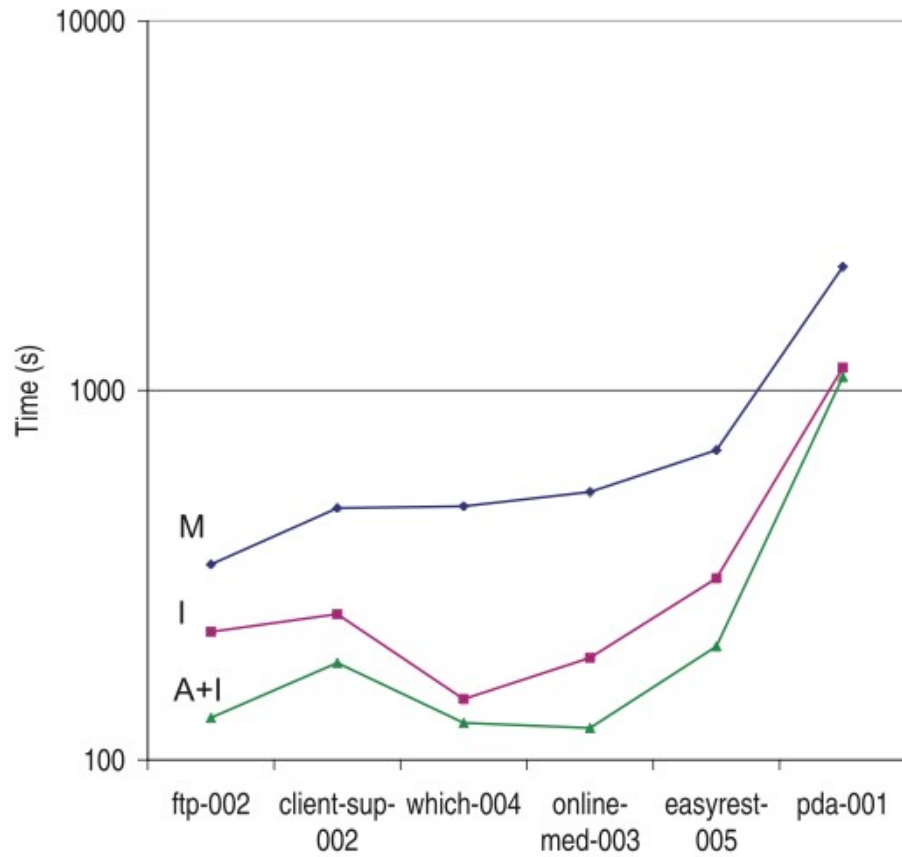


# Contract Specification Example



search ID: mbcn546

"It's a simple procedure. I'm blindfolded, spun around and then I attempt to reattach your tail."



[ Time in logarithmic scale ]

- ▣ Compared to manual contract specification:
  - ▢ The time required to design the contract is reduced by 67%
  - ▢ 77% less errors during the contract specification process
- ▣ Our proposal worked especially well in cases where functionality is not scattered across multiple small interfaces
  - ▢ The automatic contract generator works pairwise
- ▣ As regards future work we have extended the underlying formalisation with adaptation goals expressed in temporal logic and we are currently working on including security concerns to the adaptation

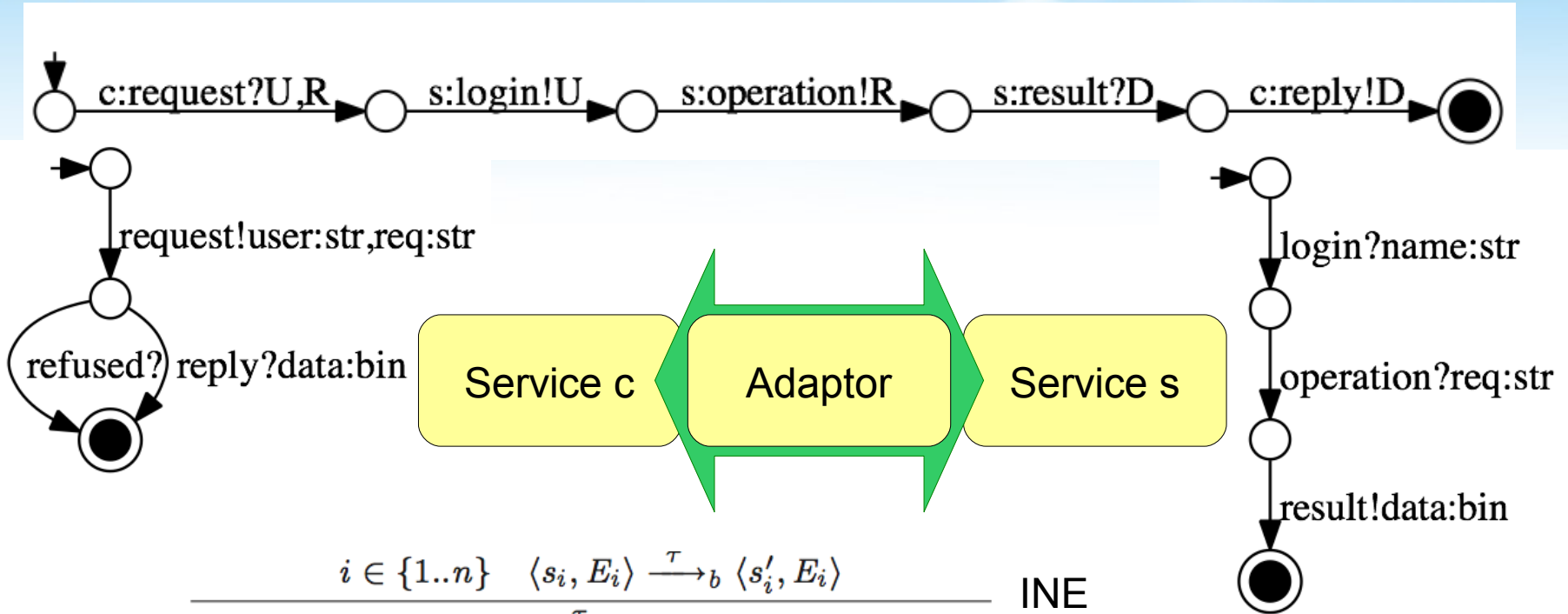




**INTERNET IS DOWN**

there's absolutely nothing to do

# Adaptation Behaviour



$$\frac{i \in \{1..n\} \quad \langle s_i, E_i \rangle \xrightarrow{\tau}_b \langle s'_i, E_i \rangle}{\{as_1, \dots, \langle s_i, E_i \rangle, \dots, as_n\} \xrightarrow{\tau}_c \{as_1, \dots, \langle s'_i, E_i \rangle, \dots, as_n\}} \quad \text{INE}$$

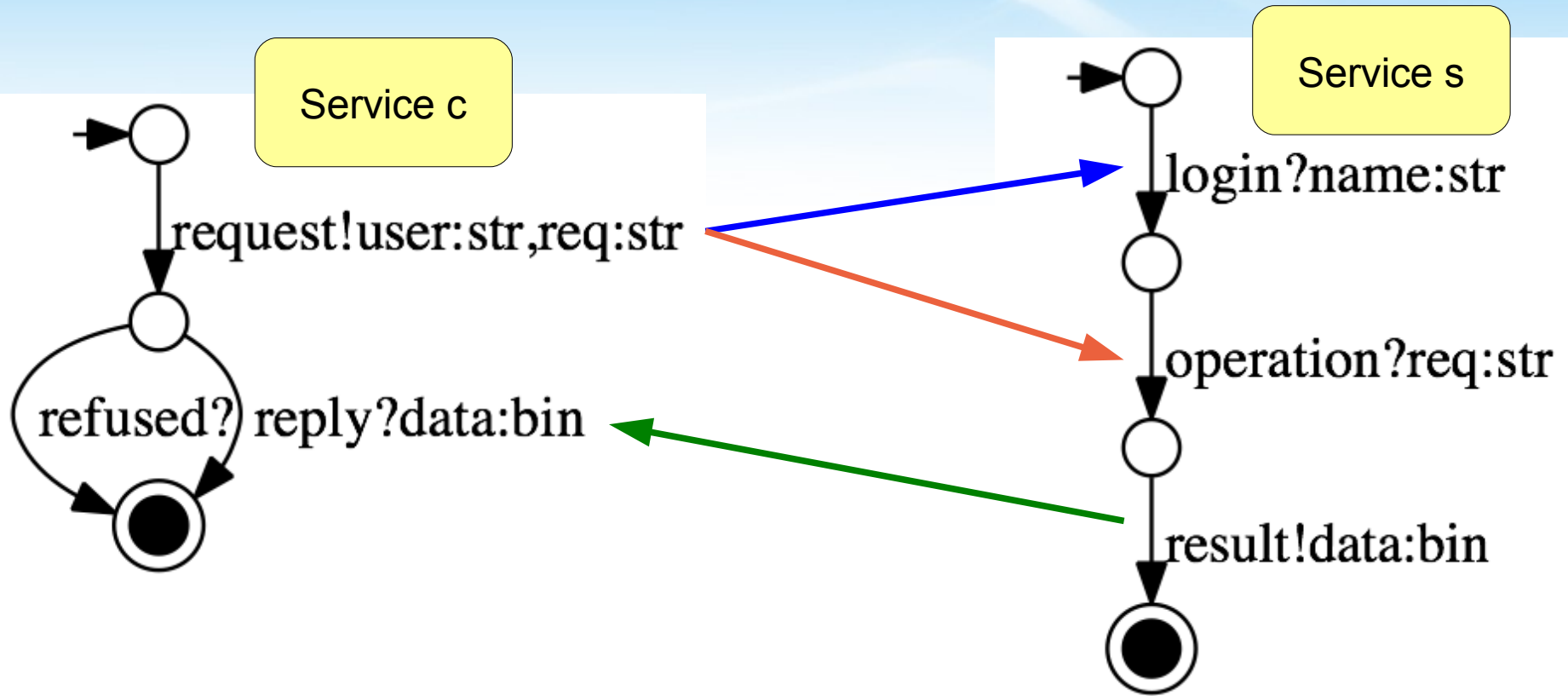
$$\begin{array}{c} i, j \in \{1..n\} \quad i \neq j \\ \langle s_i, E_i \rangle \xrightarrow{a!v}_b \langle s'_i, E_i \rangle \quad \langle s_j, E_j \rangle \xrightarrow{a?x}_b \langle s'_j, E_j \rangle \\ type(x) = type(v) \quad E'_j = E_j \circ \langle x, v \rangle \end{array}$$

$$\{as_1, \dots, \langle s_i, E_i \rangle, \dots, \langle s_j, E_j \rangle, \dots, as_n\} \xrightarrow{a!v}_c \{as_1, \dots, \langle s'_i, E_i \rangle, \dots, \langle s'_j, E'_j \rangle, \dots, as_n\}$$

COM

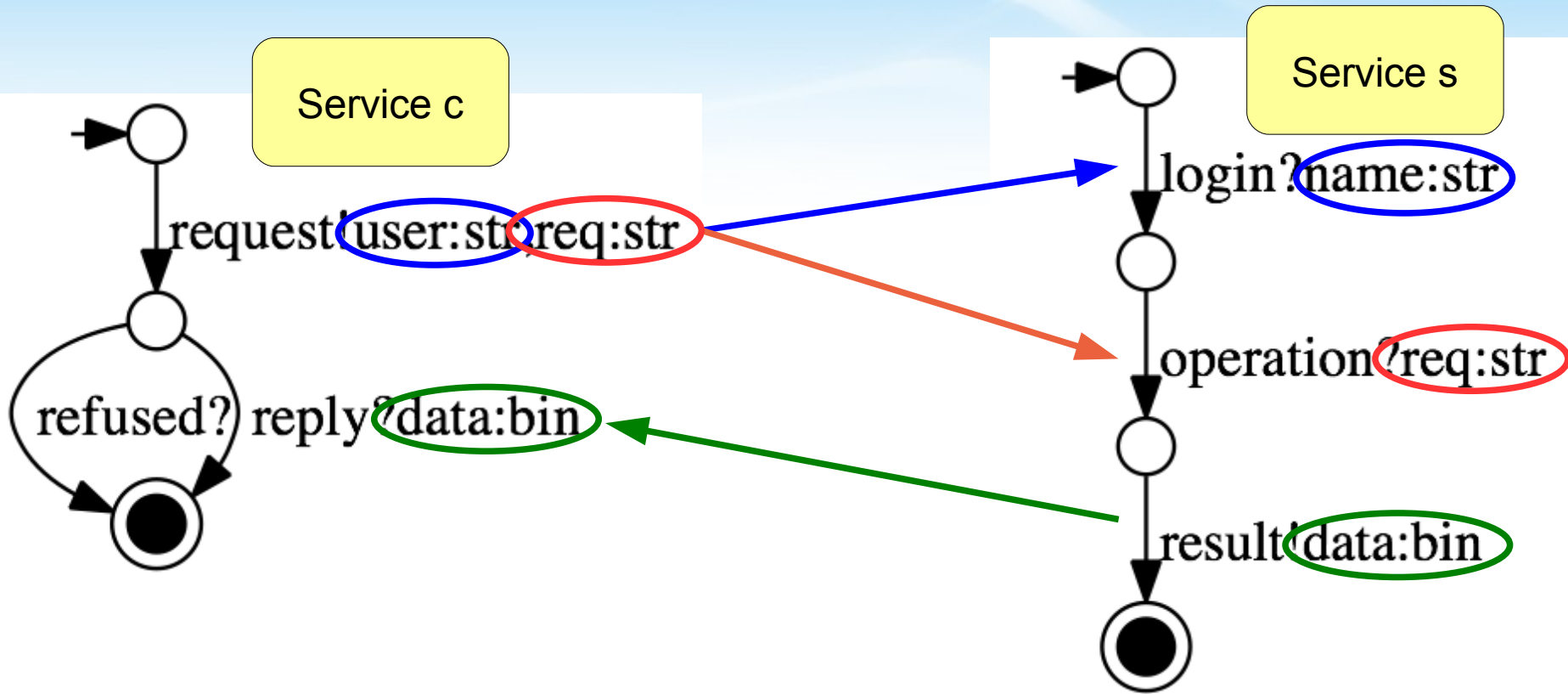
- ▶ Comparing to manual contract specification:
  - ▶ The time required to design the contract is reduced by 67%
  - ▶ 77% less errors during the contract specification process
- ▶ Our proposal worked especially well in cases where functionality is not scattered across multiple small interfaces
  - ▶ The automatic contract generator works pairwise
  - ▶
  - ▶ Y así mas...
    - ▶ Y este de tercer nivel
      - ▶ Y este de cuarto
        - ▶ Y finalmente este de quinto (y último)
    - ▶ Segundo
  - ▶ Primero
    - ▶ Y ya está
  - ▶ .....

# N to M matching



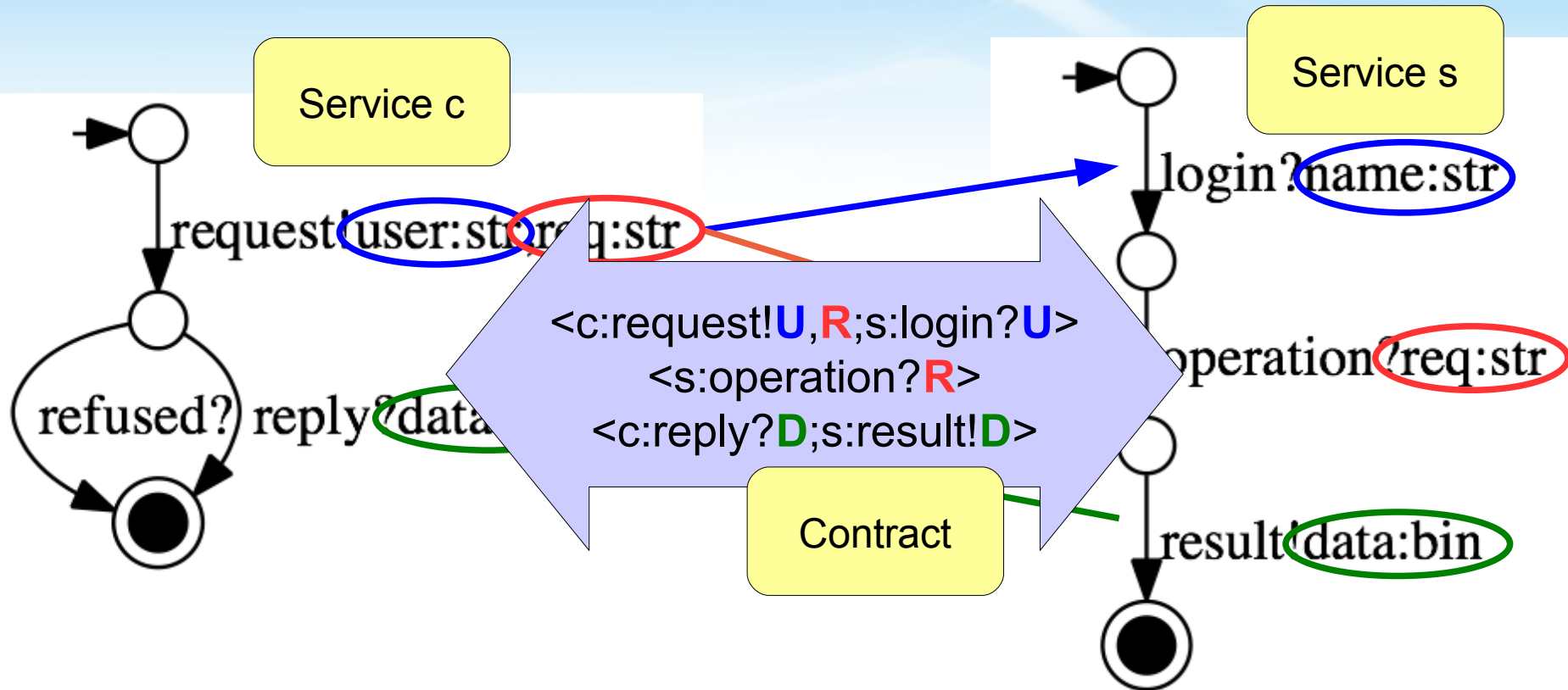
**Definition** [STS] A Symbolic Transition System or STS is a tuple  $(A, S, I, F, T)$  where:  $A$  is an alphabet which corresponds to the set of labels associated to transitions,  $S$  is a set of states,  $I \in S$  is the initial state,  $F \subseteq S$  are the final states, and  $T : S \times A \times S$  is the transition function.

# Parameter Adaptation



**Definition** [STS] A Symbolic Transition System or STS is a tuple  $(A, S, I, F, T)$  where:  $A$  is an alphabet which corresponds to the set of labels associated to transitions,  $S$  is a set of states,  $I \in S$  is the initial state,  $F \subseteq S$  are the final states, and  $T : S \times A \times S$  is the transition function.

# Adaptation Contract



**Definition** [Vector] A *vector* for a set of service  $STS_i = (A_i, S_i, I_i, F_i, T_i), i \in \{1, \dots, n\}$  is an element of  $A_j \cup (A_j \times A_k)$  with  $j, k \in \{1, \dots, n\}, j \neq k$ . Such a vector is noted  $\langle s_j:l \rangle$ , or  $\langle s_j:l, s_k:l' \rangle$  where  $s_j, s_k$  are service identifiers, and  $l, l'$  are labels on the alphabets of services  $A_j, A_k$ , where message parameters are substituted by placeholders relating the arguments.

# Adaptation Behaviour

