Compositionality of Safe Communication in Systems of Team Automata

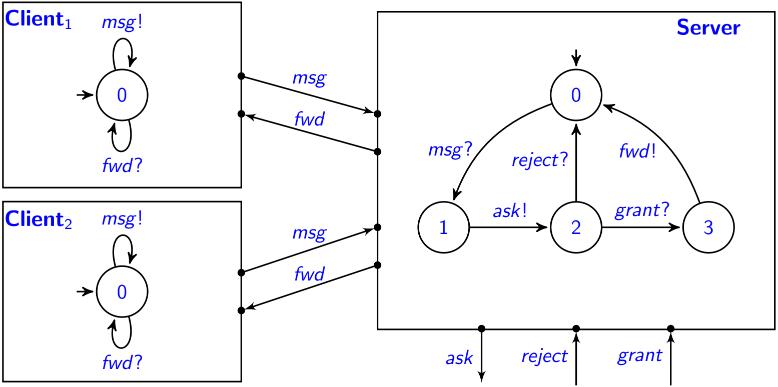
Maurice H. ter Beek Rolf Hennicker Jetty Kleijn

ISTI–CNR, Pisa, Italy

LMU Munich, Germany

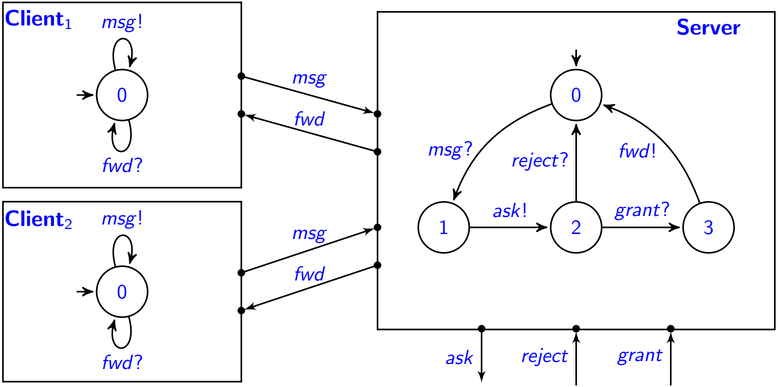
LIACS, Leiden University, The Netherlands

We consider: Systems of Communicating Components



Communicating actions $Com(\mathcal{S}) = \{msg, fwd\}$

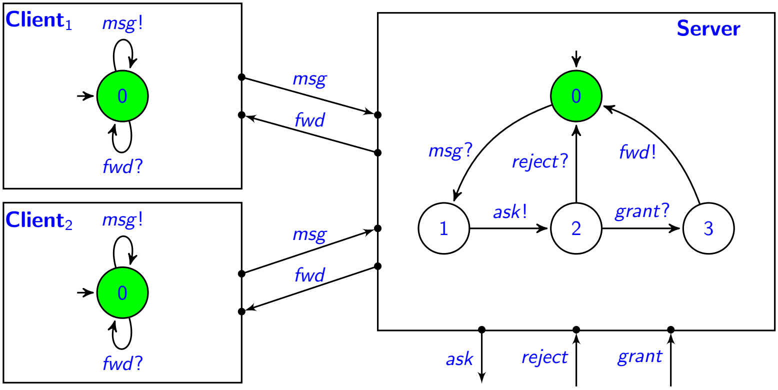
We consider: Systems of Communicating Components



Communicating actions $Com(\mathcal{S}) = \{msg, fwd\}$

System transition: simultaneous execution of a communicating action. In principle, any number of components can participate.

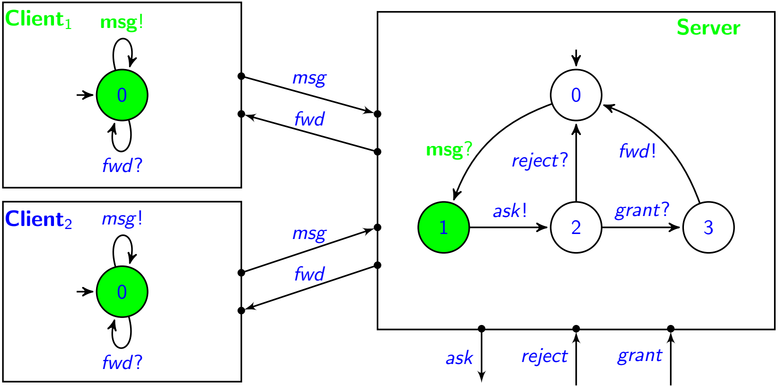
We consider: Systems of Communicating Components



Communicating actions $Com(\mathcal{S}) = \{msg, fwd\}$

System transition: simultaneous execution of a communicating action. In principle, any number of components can participate.

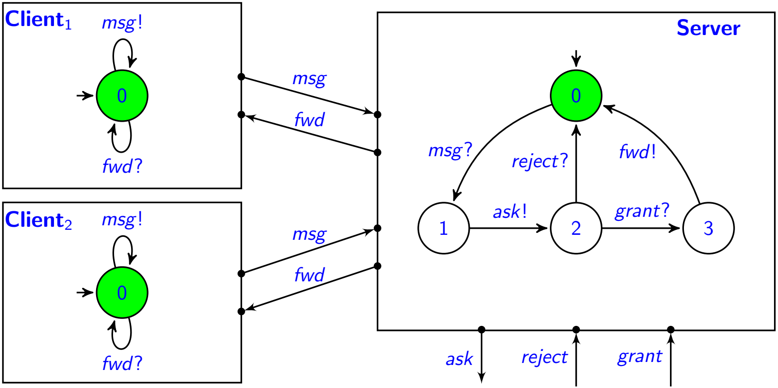
We consider: Systems of Communicating Components



Communicating actions $\mathit{Com}(\mathcal{S}) = \{\mathit{msg}, \mathit{fwd}\}$

System transition: simultaneous execution of a communicating action. In principle, any number of components can participate.

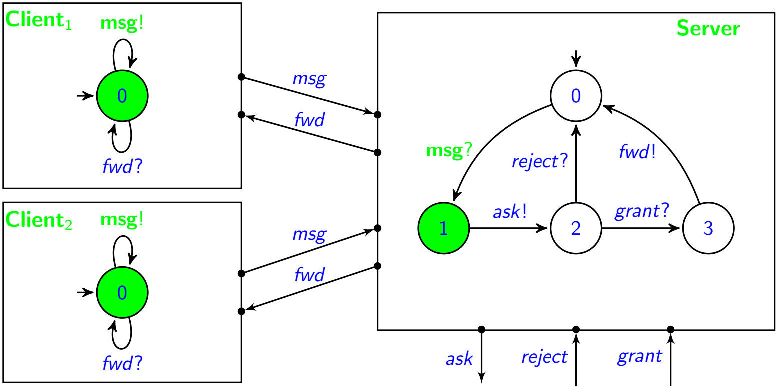
We consider: Systems of Communicating Components



Communicating actions $Com(\mathcal{S}) = \{msg, fwd\}$

System transition: simultaneous execution of a communicating action. In principle, any number of components can participate.

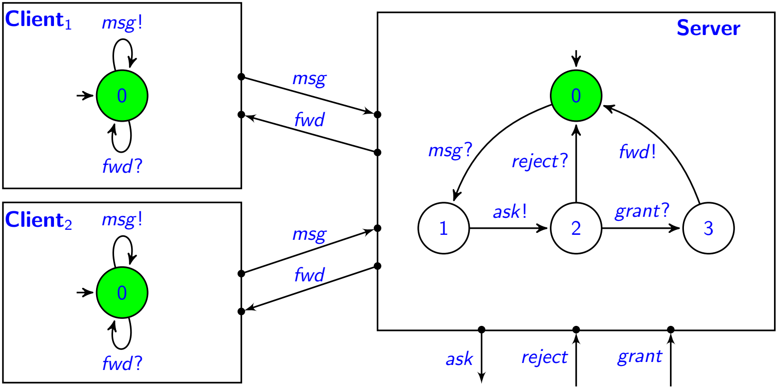
We consider: Systems of Communicating Components



Communicating actions $Com(\mathcal{S}) = \{msg, fwd\}$

System transition: simultaneous execution of a communicating action. In principle, any number of components can participate.

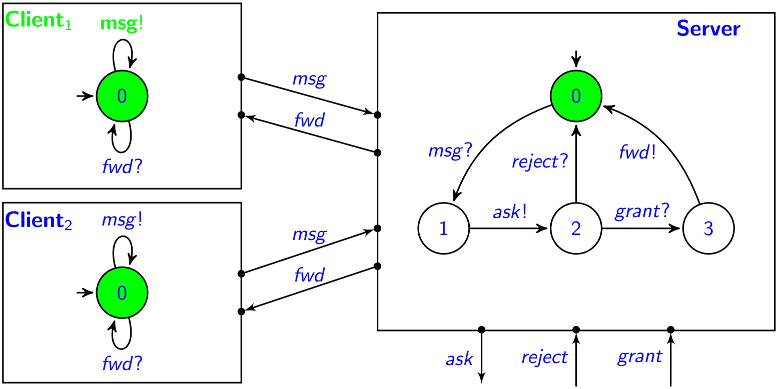
We consider: Systems of Communicating Components



Communicating actions $Com(\mathcal{S}) = \{msg, fwd\}$

System transition: simultaneous execution of a communicating action. In principle, any number of components can participate.

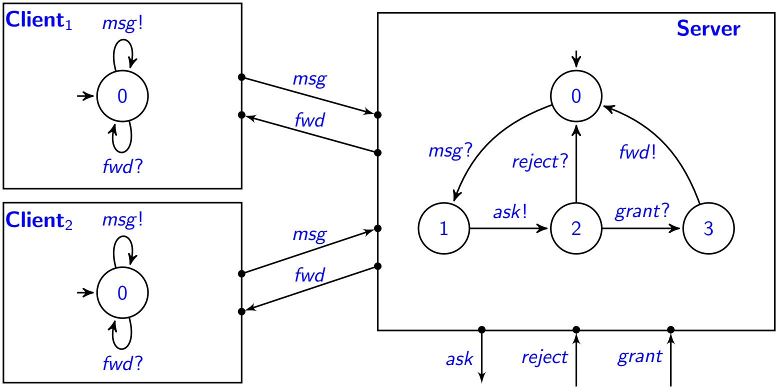
We consider: Systems of Communicating Components



Communicating actions $Com(\mathcal{S}) = \{msg, fwd\}$

System transition: simultaneous execution of a communicating action. In principle, any number of components can participate.

We consider: Systems of Communicating Components

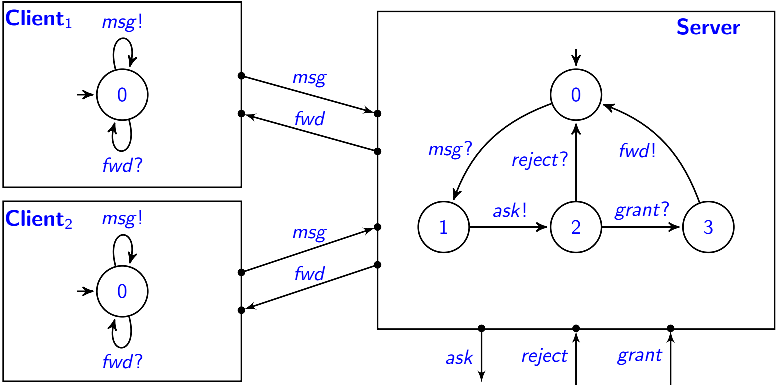


Communicating actions: $Com(\mathcal{S}) = \{msg, fwd\}$

System transition: simultaneous execution of a communicating action. In principle, any number of components can participate.

Not all system transtions are meaningful!

We consider: Systems of Communicating Components



Idea: Specify for each communicating action a a *synchronisation type* $st(a)$; e.g. $st(msg) = 1 \rightarrow 1$, $st(fwd) = 1 \rightarrow *$.
This generates a set of system transitions formalised as an extended team automaton $\mathcal{T}(st)$. It has transitions like

$$(0,0,3) \xrightarrow{(\{\text{Server}\},fwd,\{\text{Client}_1,\text{Client}_2\})} (0,0,0)$$

Our Contributions

- Specification of teams through individual synchronisation types *per action*; in general $[\mathtt{min}_{\mathtt{out}}, \mathtt{max}_{\mathtt{out}}] \rightarrow [\mathtt{min}_{\mathtt{in}}, \mathtt{max}_{\mathtt{in}}]$
  (peer-to-peer, multicast, broadcast, gathering, master-slave, ...)
- Study of *communication-safety properties* in dependence of synchronisation type specifications
  $\rightarrow$ *receptiveness, responsiveness*
- *Composition* of systems and criteria for *preservation* of communication-safety properties after composition
  $\rightarrow$ compositionality results!

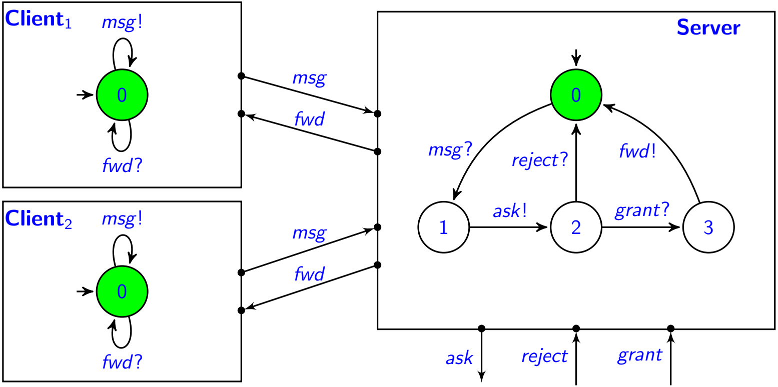
Our Contributions

- Specification of teams through individual synchronisation types *per action*; in general $[\mathtt{min}_{\mathtt{out}}, \mathtt{max}_{\mathtt{out}}] \to [\mathtt{min}_{\mathtt{in}}, \mathtt{max}_{\mathtt{in}}]$
  (peer-to-peer, multicast, broadcast, gathering, master-slave, ...)

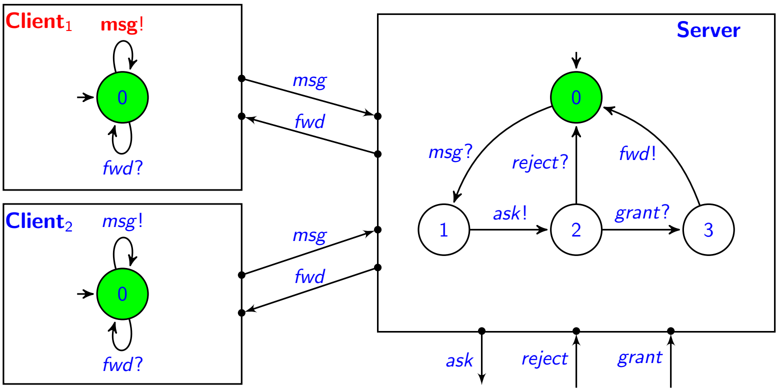
- Study of *communication-safety properties* in dependence of synchronisation type specifications
  $\to$ *receptiveness, responsiveness*

- *Composition* of systems and criteria for *preservation* of communication-safety properties after composition
  $\to$ compositionality results!

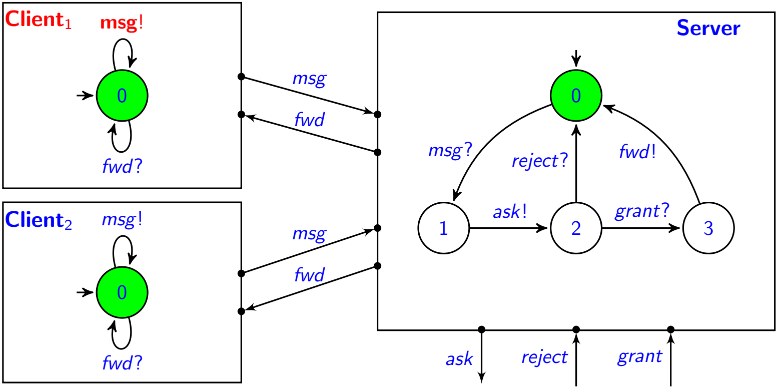
On Safe Communication: Receptiveness



On Safe Communication: Receptiveness



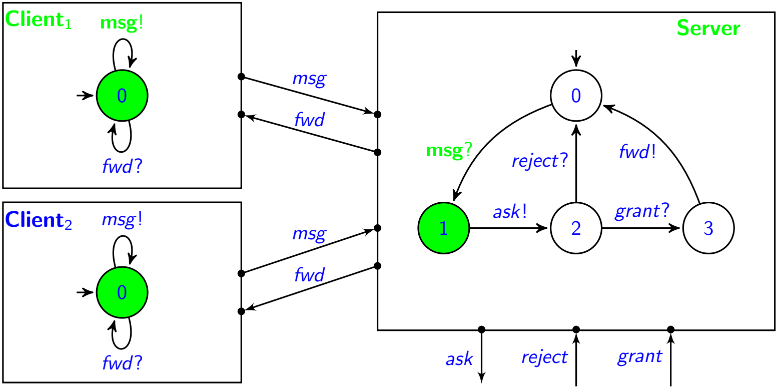
On Safe Communication: Receptiveness



Receptiveness requirement:

$$\texttt{rcp}(\{\text{Client}_1\}, msg)@(0,0,0)$$

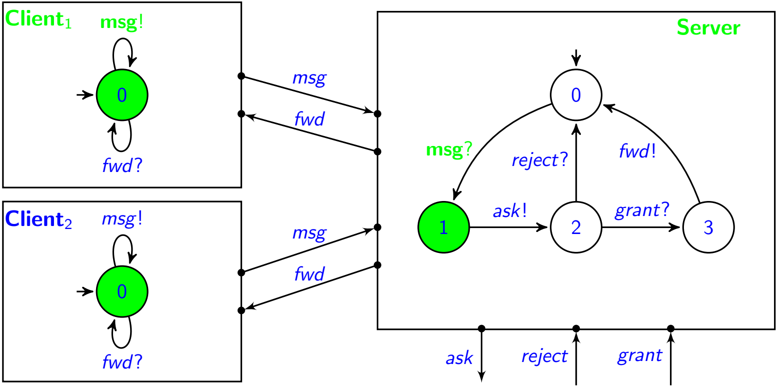
On Safe Communication: Receptiveness



Receptiveness requirement:

$\texttt{rcp}(\{\text{Client}_1\}, \mathit{msg})@(0, 0, 0)$

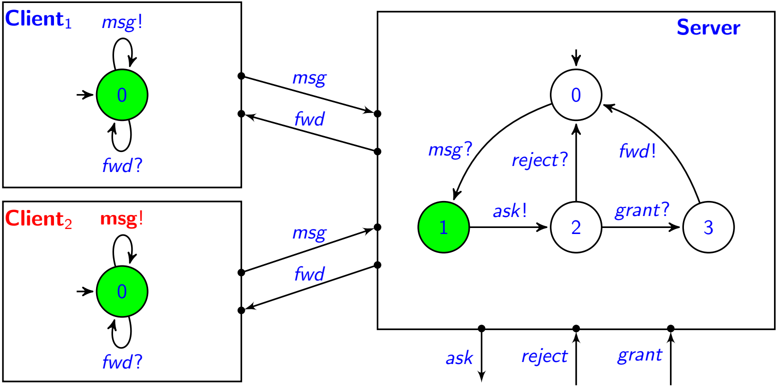
On Safe Communication: Receptiveness



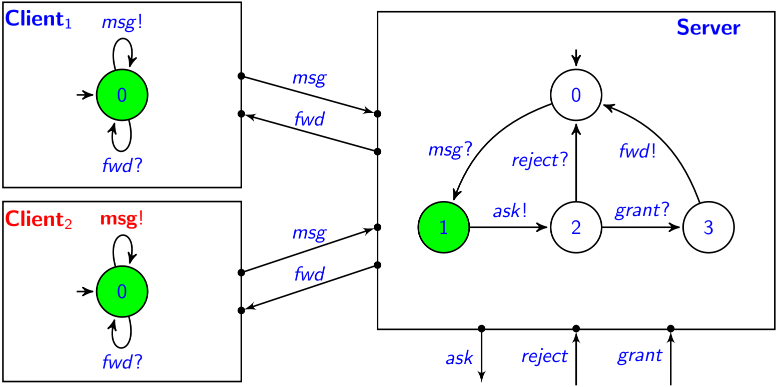
Receptiveness requirement:

$\texttt{rcp}(\{\text{Client}_1\}, \mathit{msg})@(0,0,0)$ ✓ $\mathcal{T}(st)$ is “strongly receptive”

On Safe Communication: Receptiveness



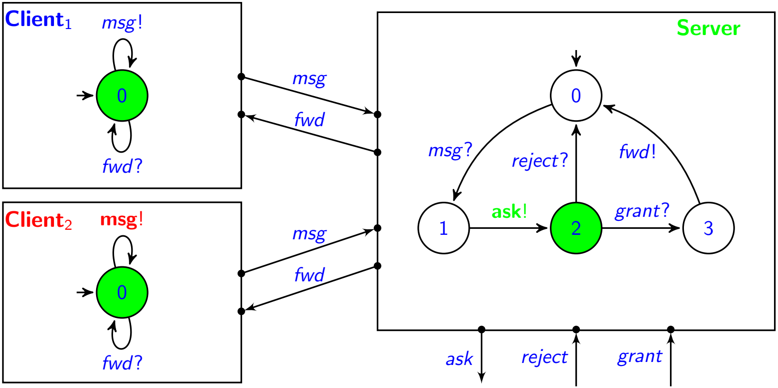
On Safe Communication: Receptiveness



Receptiveness requirement:

$\mathtt{rcp}(\{\mathsf{Client}_2\}, \mathit{msg})@(0, 0, 1)$

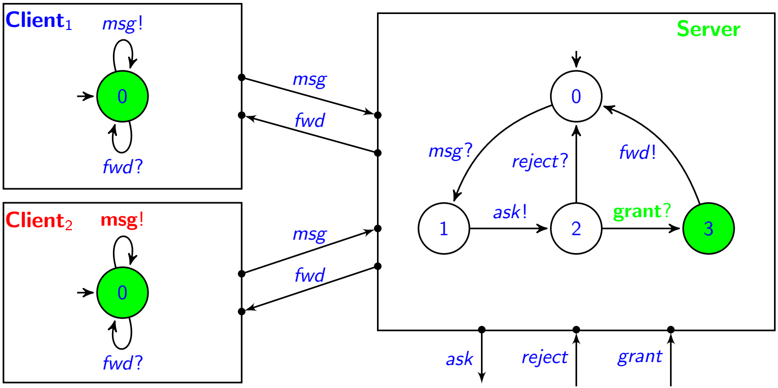
On Safe Communication: Receptiveness



Receptiveness requirement:

$\texttt{rcp}(\{\text{Client}_2\}, \mathit{msg})@(0, 0, 1)$

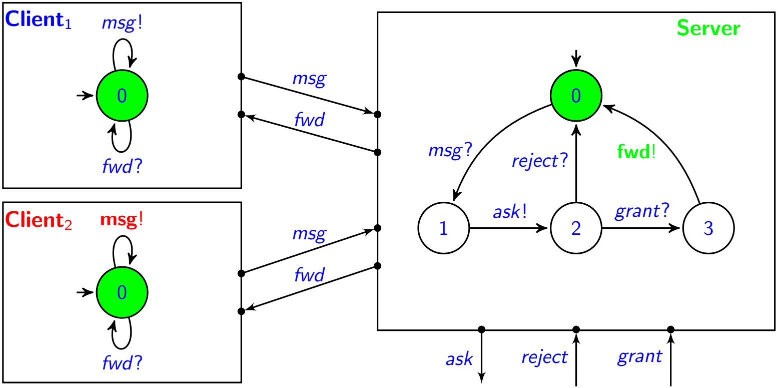
On Safe Communication: Receptiveness



Receptiveness requirement:

$\mathtt{rcp}(\{\mathrm{Client}_2\}, \mathit{msg})@(0, 0, 1)$

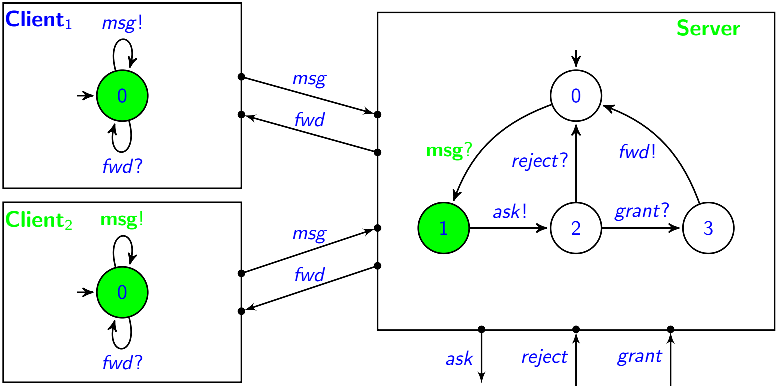
On Safe Communication: Receptiveness



Receptiveness requirement:

$\texttt{rcp}(\{\text{Client}_2\}, \mathit{msg})@(0, 0, 1)$

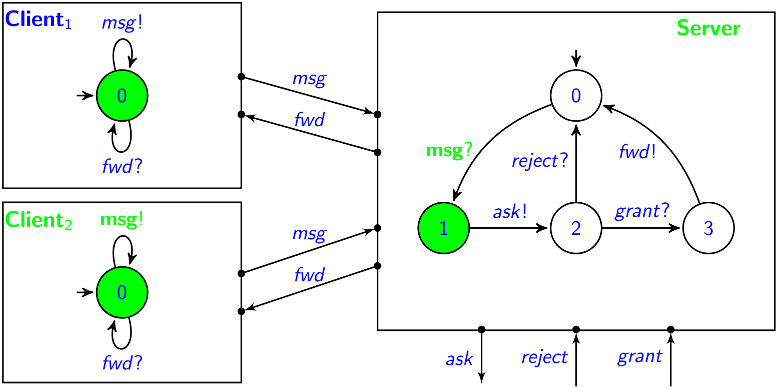
On Safe Communication: Receptiveness



Receptiveness requirement:

$\texttt{rcp}(\{\text{Client}_2\}, \mathit{msg})@(0, 0, 1)$

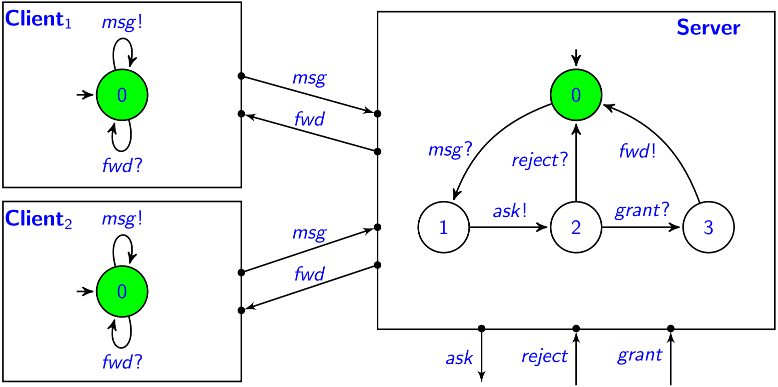
On Safe Communication: Receptiveness



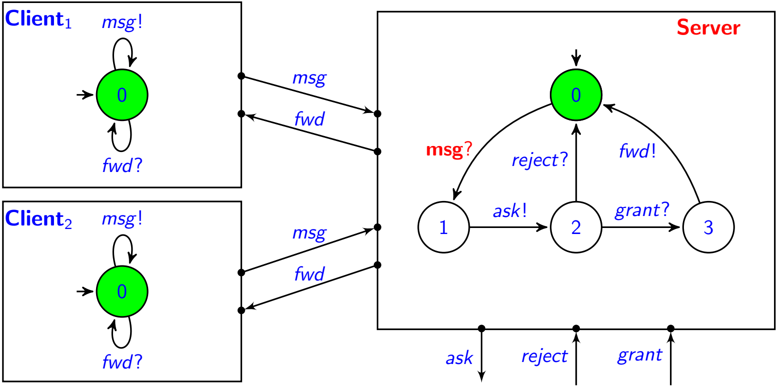
Receptiveness requirement:

$\texttt{rcp}(\{\text{Client}_2\}, \mathit{msg})@(0,0,1)$ ✓ $\mathcal{T}(st)$ is "weakly receptive"

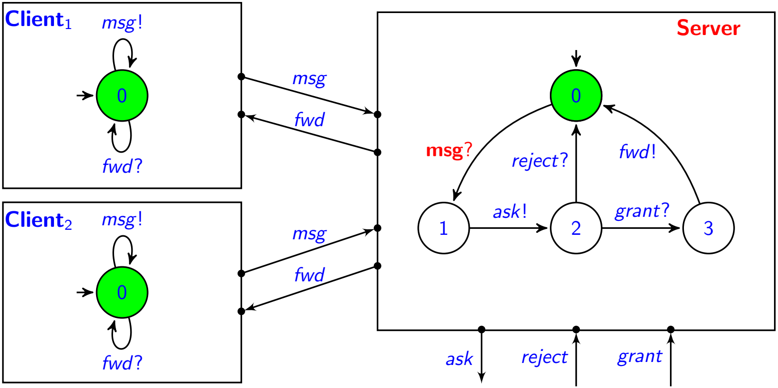
On Safe Communication: Responsiveness



On Safe Communication: Responsiveness



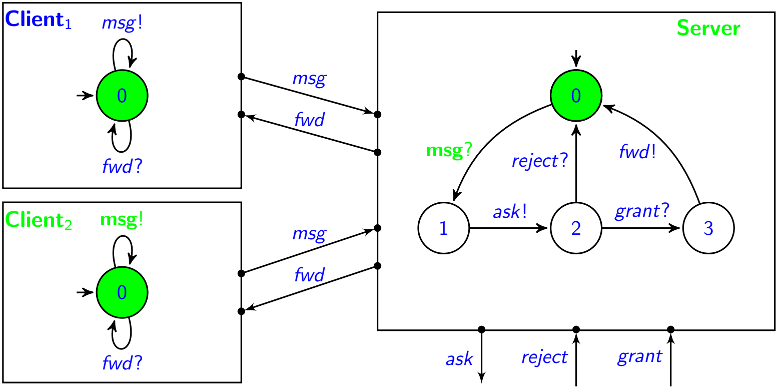
On Safe Communication: Responsiveness



Responsiveness requirement:

$\texttt{rsp}(\{\text{Server}\}, \mathit{msg})@(0, 0, 0)$

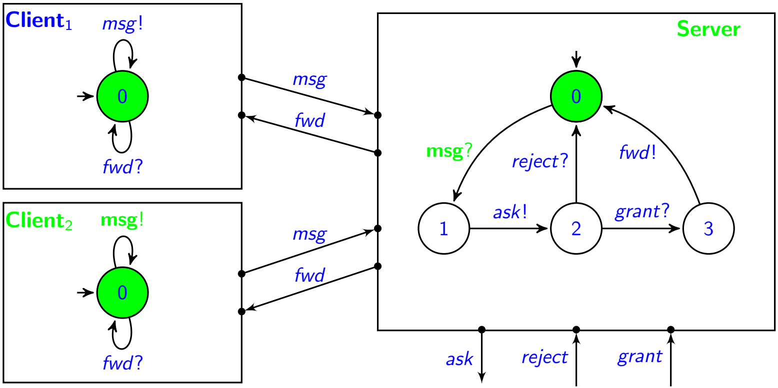
On Safe Communication: Responsiveness



Responsiveness requirement:

$\texttt{rsp}(\{\text{Server}\}, \mathit{msg})@(0,0,0)$

On Safe Communication: Responsiveness



Responsiveness requirement:

$\texttt{rsp}(\{\text{Server}\}, \mathit{msg})@(0,0,0)$ ✓ $\mathcal{T}(st)$ is "strongly responsive"

Communication-Safety

General idea: A team $\mathcal{T}(st)$ satisfies a communication requirement (receptiveness, responsiveness) if whenever a group of components in the team issues a request for communication it can successfully find partners to join.

- If the partners join immediately the team $\mathcal{T}(st)$ is *strongly receptive* (*strongly responsive*, resp.).
- If the partners join after execution of some intermediate actions the team $\mathcal{T}(st)$ is *weakly receptive* (*weakly responsive*)
- The team $\mathcal{T}(st)$ is **strongly communication-safe** if it is strongly receptive and strongly responsive.
- It is **weakly communication-safe** if it is weakly receptive and weakly responsive.

Comparison with the Literature

- *Receptiveness in synchronous systems:*
  [de Alfaro, Henzinger 2001], [Larsen, Nyman, Wasowski 2007], [Lüttgen, Vogler, Fendrich 2015], ...
- *Responsiveness in synchronous systems:*
  [Carmona, Cortadella 2002], [Carrez,Fantechi,Najm 2003], [Durán,Ouederni,Salaün 2012]

Comparison with the Literature

- *Receptiveness in synchronous systems:*
  [de Alfaro, Henzinger 2001], [Larsen, Nyman, Wasowski 2007], [Lüttgen, Vogler, Fendrich 2015], ...
- *Responsiveness in synchronous systems:*
  [Carmona, Cortadella 2002], [Carrez,Fantechi,Najm 2003], [Durán,Ouederni,Salaün 2012]

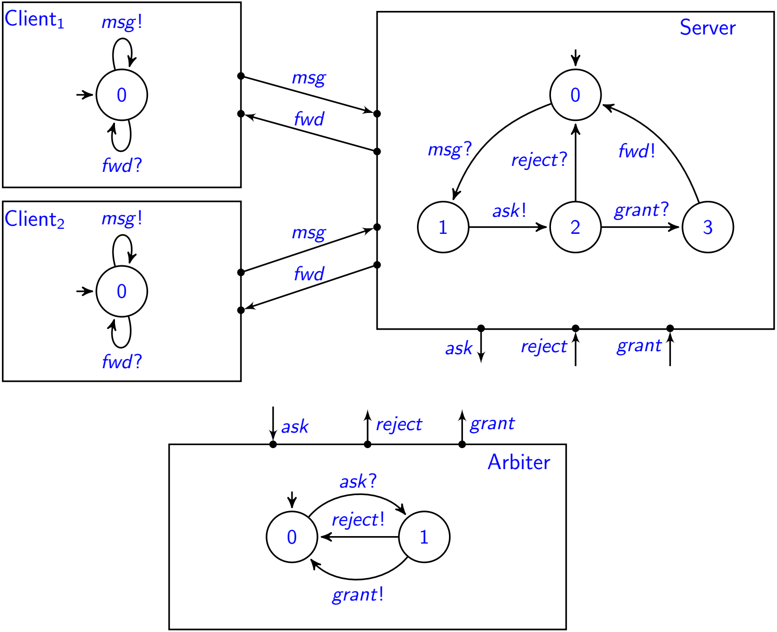
The above approaches are for systems, in which actions follow a one-to-one synchronisation style.

Our approach supports any kind of synchronisation type individually determined per action (thus generalising [ter Beek, Carmona, Hennicker, Kleijn 2017]).

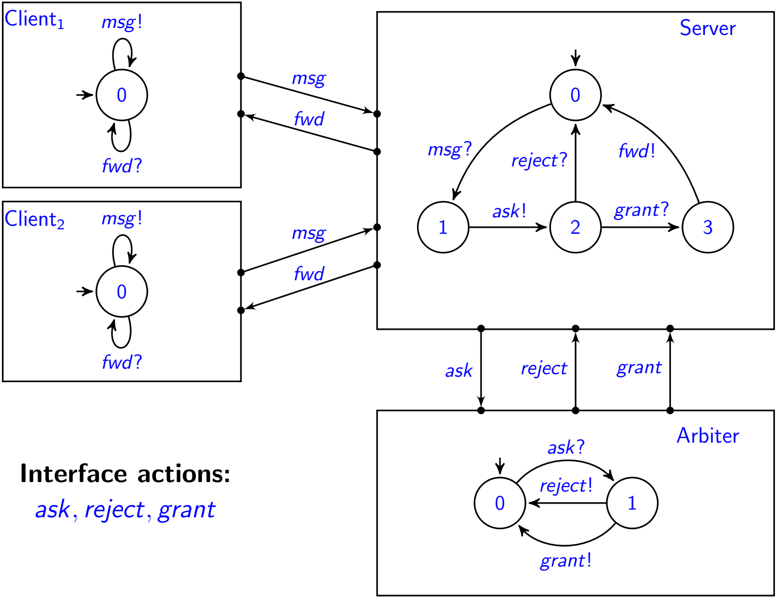
We also support weak notions of receptiveness and responsiveness.

... and now there come some compostionality results

System Composition: Example



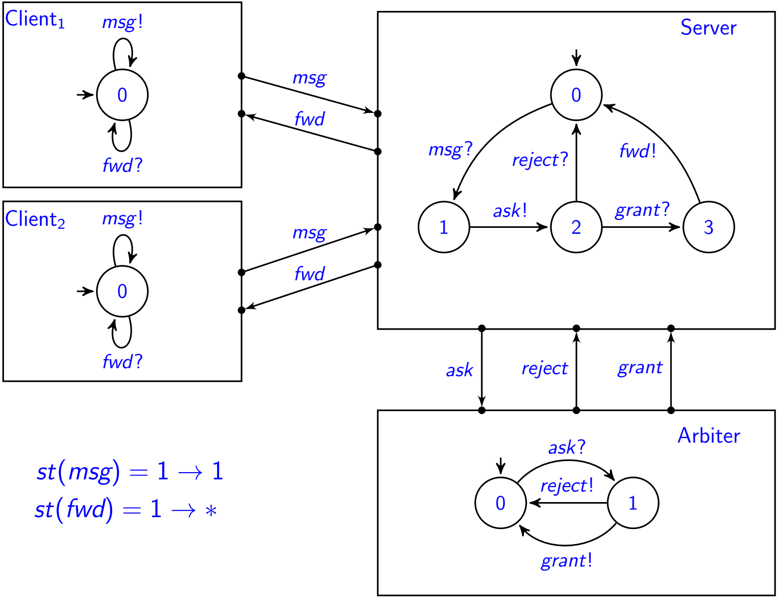
System Composition: Example



Interface actions:

ask, *reject*, *grant*

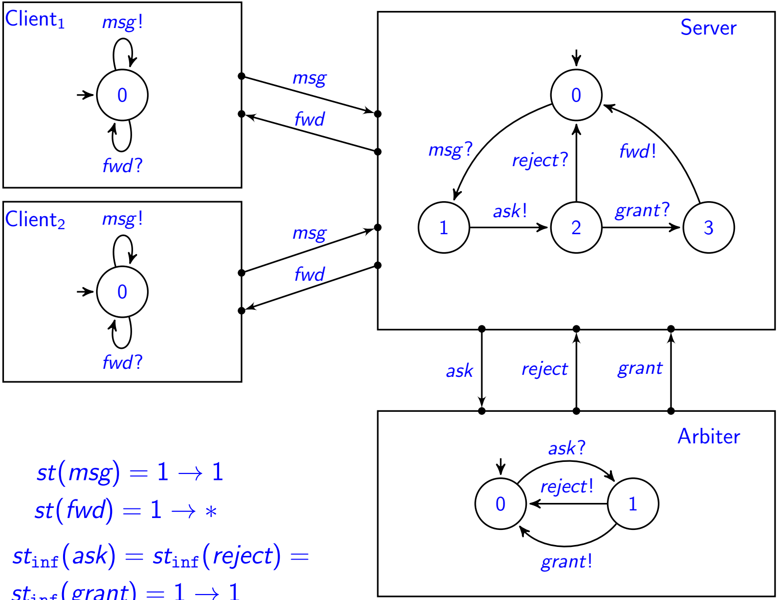
Synchronisation Type Specifications: Example



$st(msg) = 1 \to 1$

$st(fwd) = 1 \to *$

Synchronisation Type Specifications: Example



$$st(msg) = 1 \to 1$$
$$st(fwd) = 1 \to *$$
$$st_{\mathtt{inf}}(ask) = st_{\mathtt{inf}}(reject) =$$
$$st_{\mathtt{inf}}(grant) = 1 \to 1$$

System Composition: General Definitions

Let $\mathcal{S}_1 = \{\mathcal{A}_1, \ldots, \mathcal{A}_k\}$ and $\mathcal{S}_2 = \{\mathcal{B}_1, \ldots, \mathcal{B}_m\}$ be two component systems (more generally, $\mathcal{S}_1, \ldots, \mathcal{S}_n$).

- $\mathcal{S}_1$ and $\mathcal{S}_2$ are *composable* if $Com(\mathcal{S}_1) \cap Com(\mathcal{S}_2) = \emptyset$
- The *composition* of $\mathcal{S}_1$ and $\mathcal{S}_2$ is the system
  $$\mathcal{S}_1 \otimes \mathcal{S}_2 = \{\mathcal{A}_1, \ldots, \mathcal{A}_k, \mathcal{B}_1, \ldots, \mathcal{B}_m\}$$
- The *interface actions* of $\mathcal{S}_1 \otimes \mathcal{S}_2$ are given by
  $$Com(\mathcal{S}_1 \otimes \mathcal{S}_2) \setminus (Com(\mathcal{S}_1) \cup Com(\mathcal{S}_2))$$

Given synchronisation type specs. st_1 over $\mathcal{S}_1$ and st_2 over $\mathcal{S}_2$. Then provide a synchronisation type $st_{\mathtt{inf}}(a)$ for each interface action a (task of the system architect). Thus we get a synchronisation type specification $st_1 \otimes_{st_{\mathtt{inf}}} st_2$ over $\mathcal{S}_1 \otimes \mathcal{S}_2$.

Preservation of Communication-Safety Properties

Let $\mathcal{S}_1, \mathcal{S}_2$ as well as st_1, st_2 and $st_{\mathtt{inf}}$ be as above.

Theorem 1
If $\mathcal{T}(st_1)$ and $\mathcal{T}(st_2)$ are strongly communication-safe <u>and</u> $\mathcal{T}(st_1 \otimes_{st_{\mathtt{inf}}} st_2)$ is strongly communication-safe <u>w.r.t. all interface actions</u>, then $\mathcal{T}(st_1 \otimes_{st_{\mathtt{inf}}} st_2)$ is strongly communication-safe.

Theorem 2
If $\mathcal{T}(st_1)$ and $\mathcal{T}(st_2)$ are weakly communication-safe <u>and</u> $\mathcal{T}(st_1 \otimes_{st_{\mathtt{inf}}} st_2)$ is weakly communication-safe <u>w.r.t. all interface actions</u>, then $\mathcal{T}(st_1 \otimes_{st_{\mathtt{inf}}} st_2)$ is weakly communication-safe provided that some additional conditions are satisfied (for instance on the form of $st_{\mathtt{inf}}$).

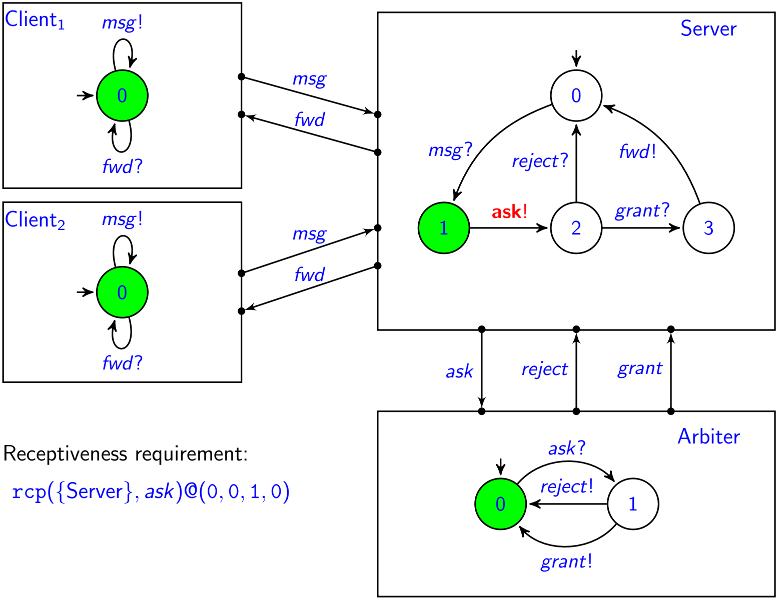
Preservation of Communication-Safety Properties

Let $\mathcal{S}_1, \mathcal{S}_2$ as well as st_1, st_2 and $st_{\mathtt{inf}}$ be as above.

Theorem 1
If $\mathcal{T}(st_1)$ and $\mathcal{T}(st_2)$ are strongly communication-safe and $\mathcal{T}(st_1 \otimes_{st_{\mathtt{inf}}} st_2)$ is strongly communication-safe w.r.t. all interface actions, then $\mathcal{T}(st_1 \otimes_{st_{\mathtt{inf}}} st_2)$ is strongly communication-safe.

Theorem 2
If $\mathcal{T}(st_1)$ and $\mathcal{T}(st_2)$ are weakly communication-safe and $\mathcal{T}(st_1 \otimes_{st_{\mathtt{inf}}} st_2)$ is weakly communication-safe w.r.t. all interface actions, then $\mathcal{T}(st_1 \otimes_{st_{\mathtt{inf}}} st_2)$ is weakly communication-safe provided that some additional conditions are satisfied (for instance on the form of $st_{\mathtt{inf}}$).

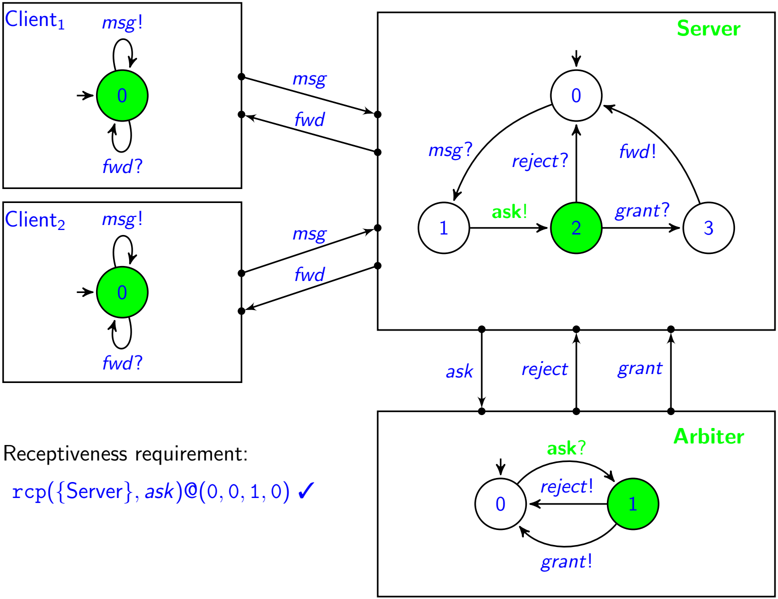
Example: Receptiveness of Interface Actions



Receptiveness requirement:

$\mathtt{rcp}(\{\text{Server}\}, \mathit{ask})@(0, 0, 1, 0)$

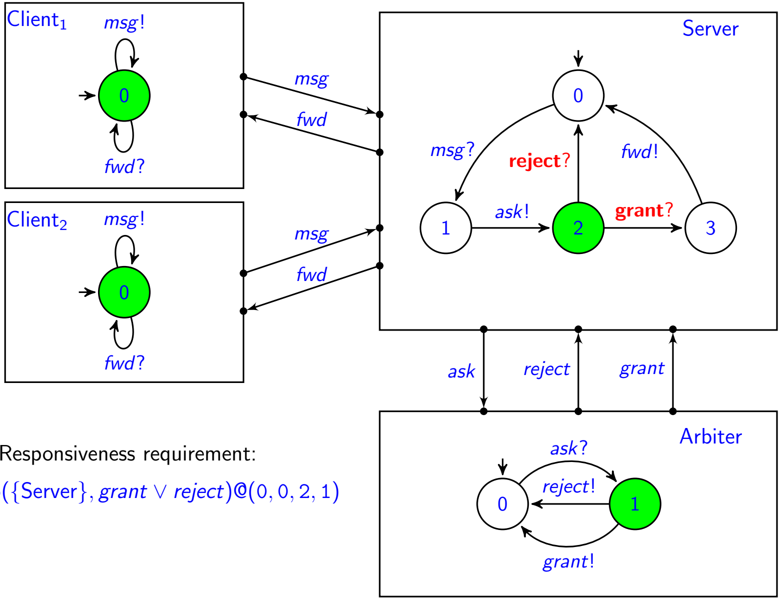
Example: Receptiveness of Interface Actions



Receptiveness requirement:

$\mathtt{rcp}(\{\text{Server}\}, \mathit{ask})@(0,0,1,0)$ ✓

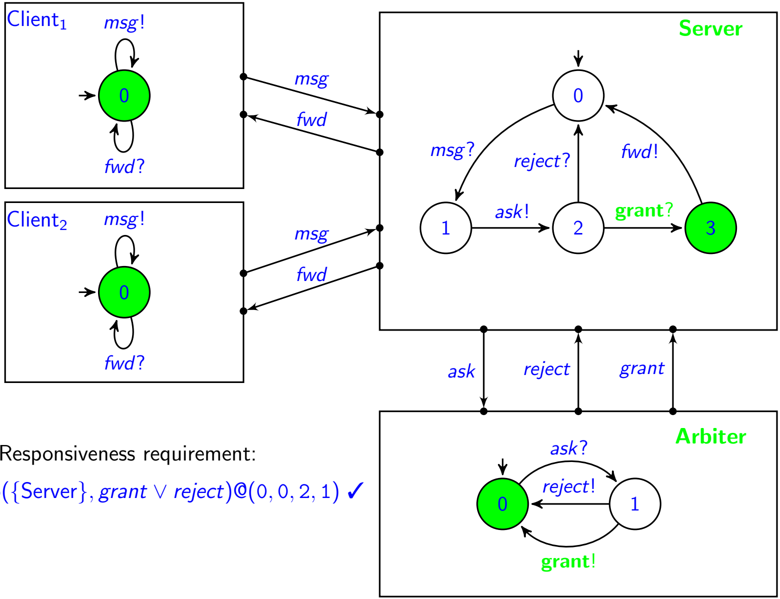
Example: Responsiveness of Interface Actions



Responsiveness requirement:

$\mathtt{rsp}(\{\text{Server}\}, \mathit{grant} \vee \mathit{reject})@(0, 0, 2, 1)$

Example: Responsiveness of Interface Actions



Responsiveness requirement:

$\texttt{rsp}(\{\text{Server}\}, \textit{grant} \vee \textit{reject})@(0, 0, 2, 1)$ ✓

Conclusion

- Generic theory for communication-safety (compatibility) in multi-component systems applicable to various kinds of synchronisation policies
- Composition of systems and synchronisation type specifications
- Compositionality results for strong and weak communication-safety
- Future research:
  - tool support for checking communication-safety properties,
  - integration into a software engineering methodology supporting encapsulation and refinement,
  - larger case studies,
  - asynchronous communication