Verification of Communicating Automata

Amrita Suresh
ENS Paris-Saclay

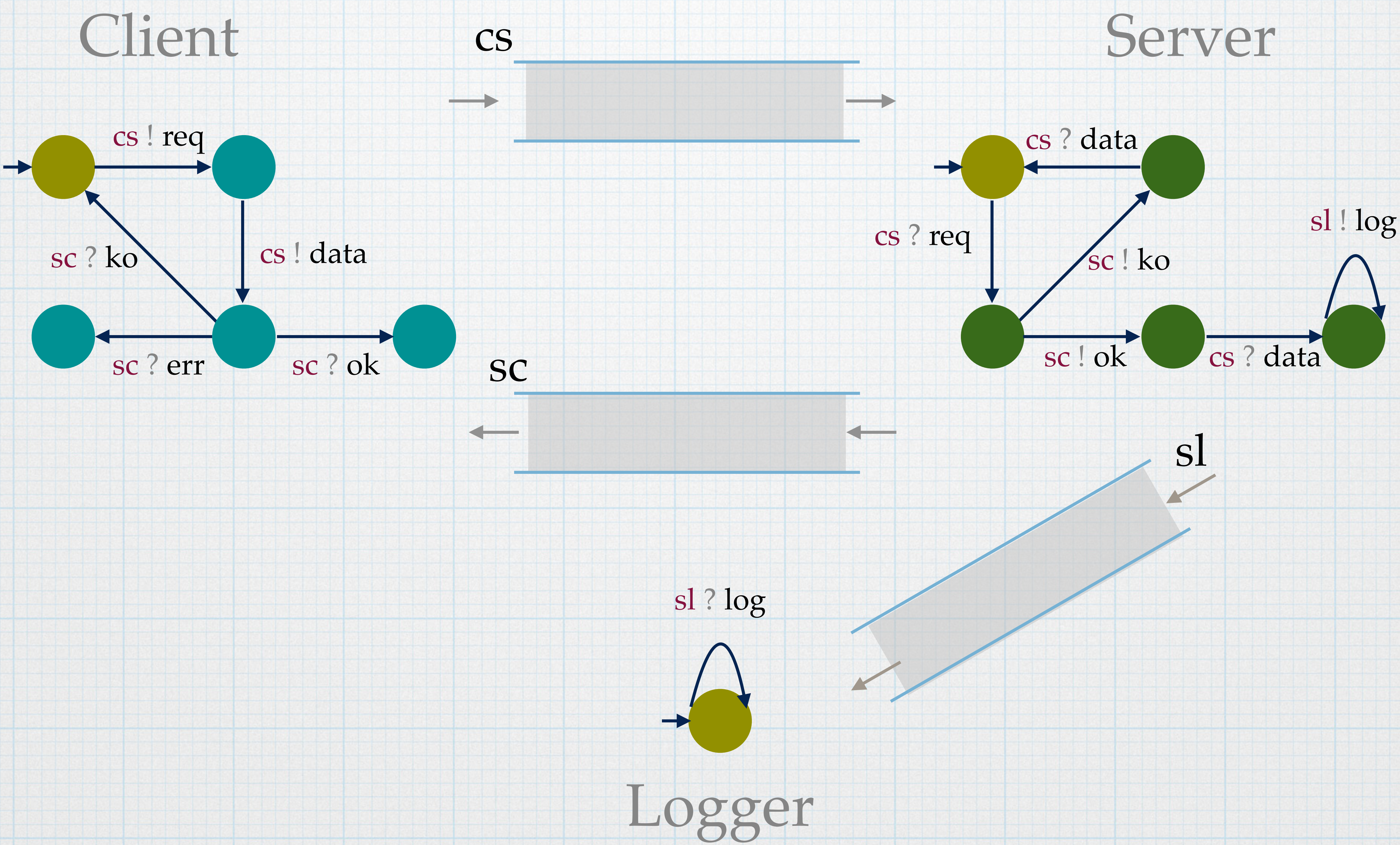
Candidate interview
21 September 2022

The model

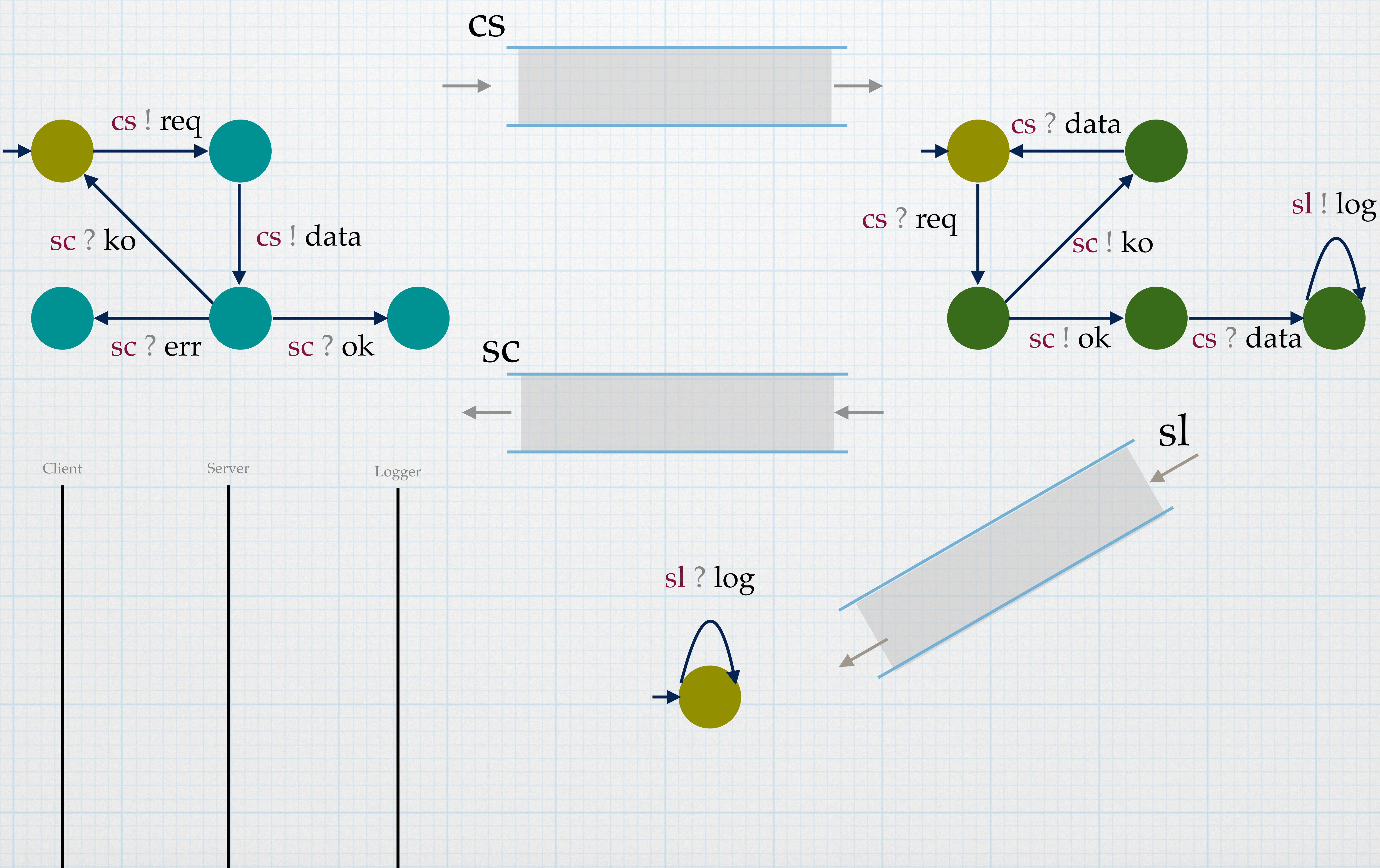
* Labelled transition systems, or automata communicating via FIFO queues
* Model distributed behaviour
    * composition of web services
    * cyber-physical devices
    * program behaviours, etc.

[LY19]

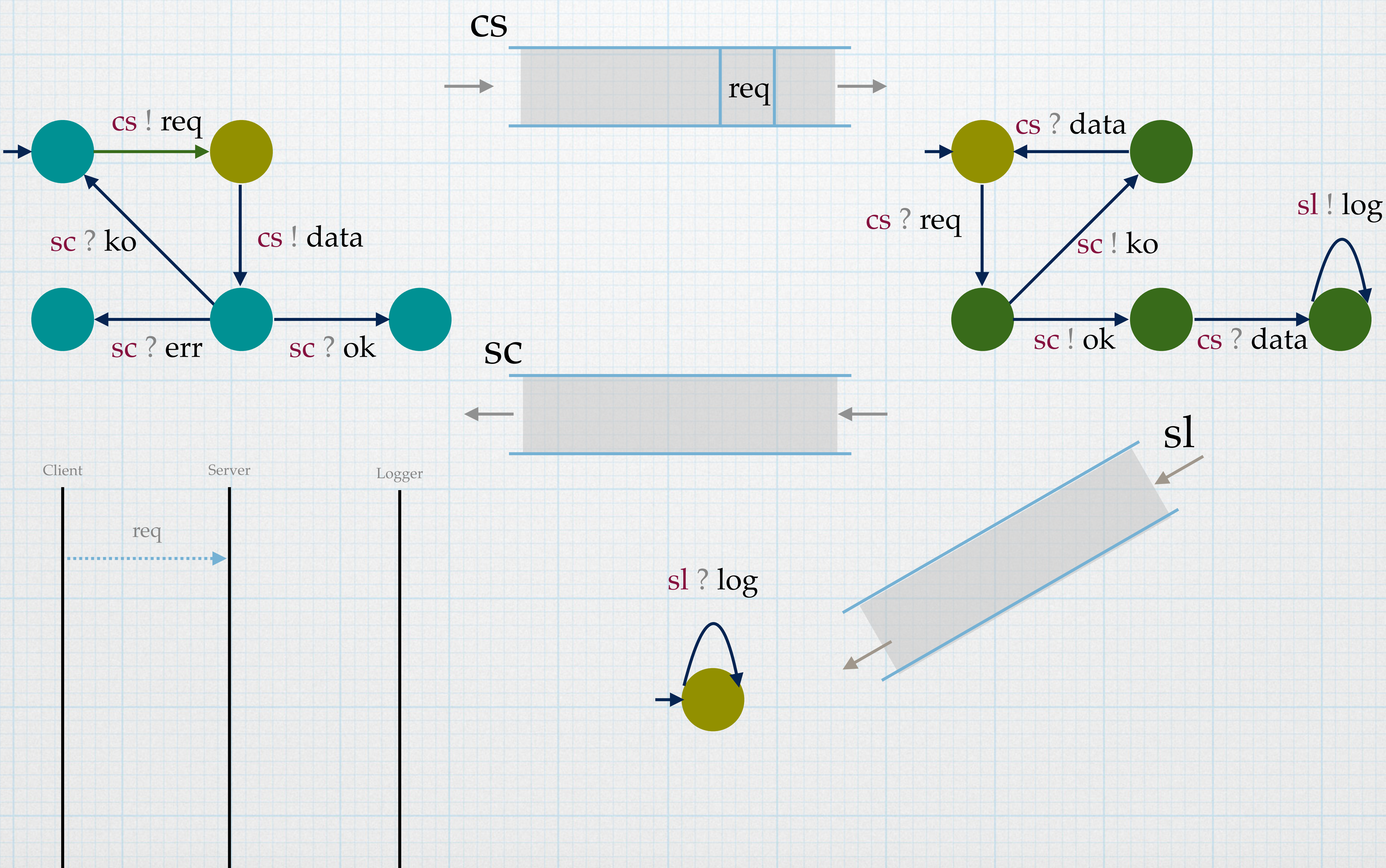
The model



The model

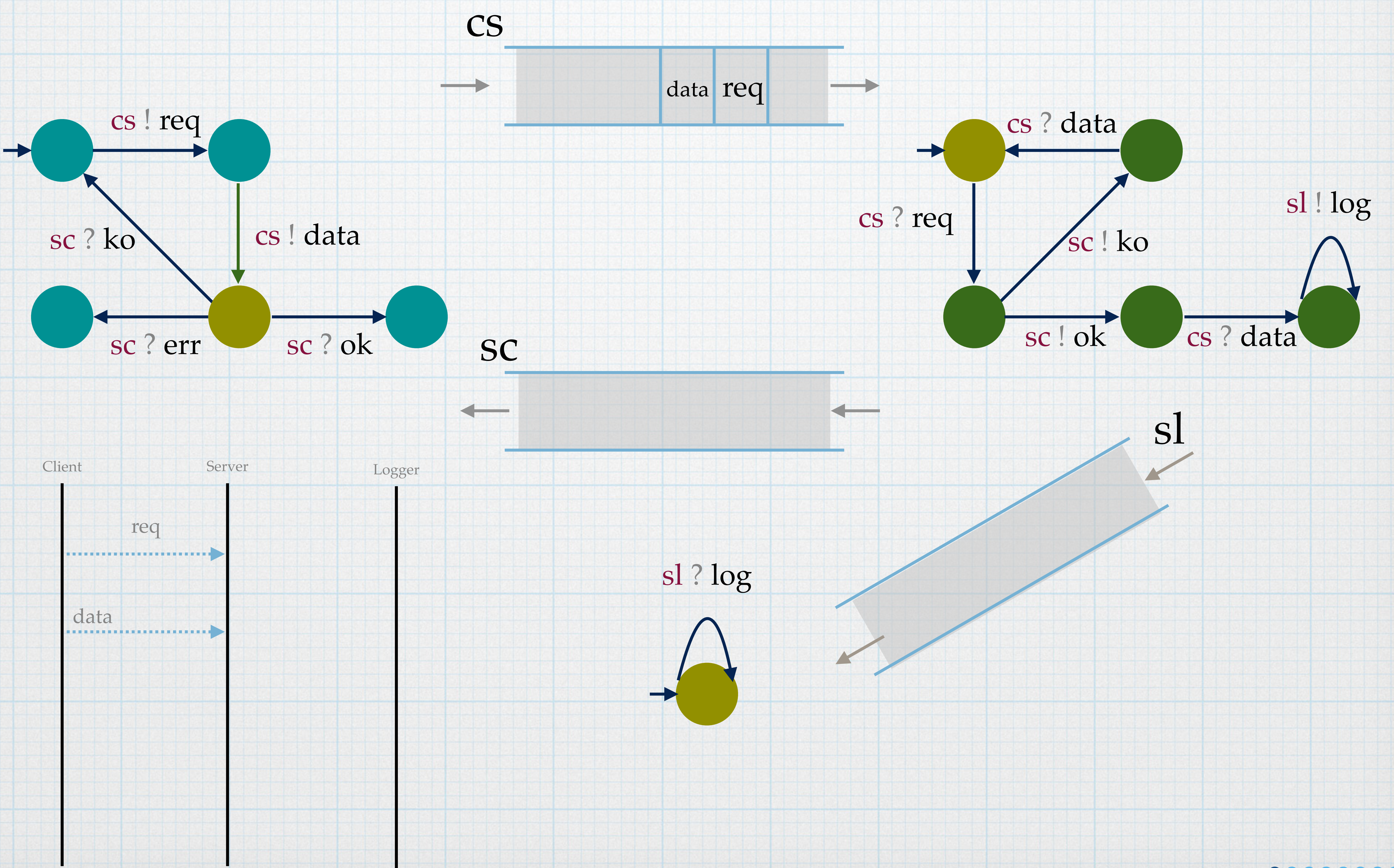

cs
sc
sl
cs ! req
cs ! data
sc ? ko
sc ? err
sc ? ok
cs ? data
cs ? req
sc ! ko
sc ! ok
cs ? data
sl ! log
sl ? log
Client
Server
Logger

The model



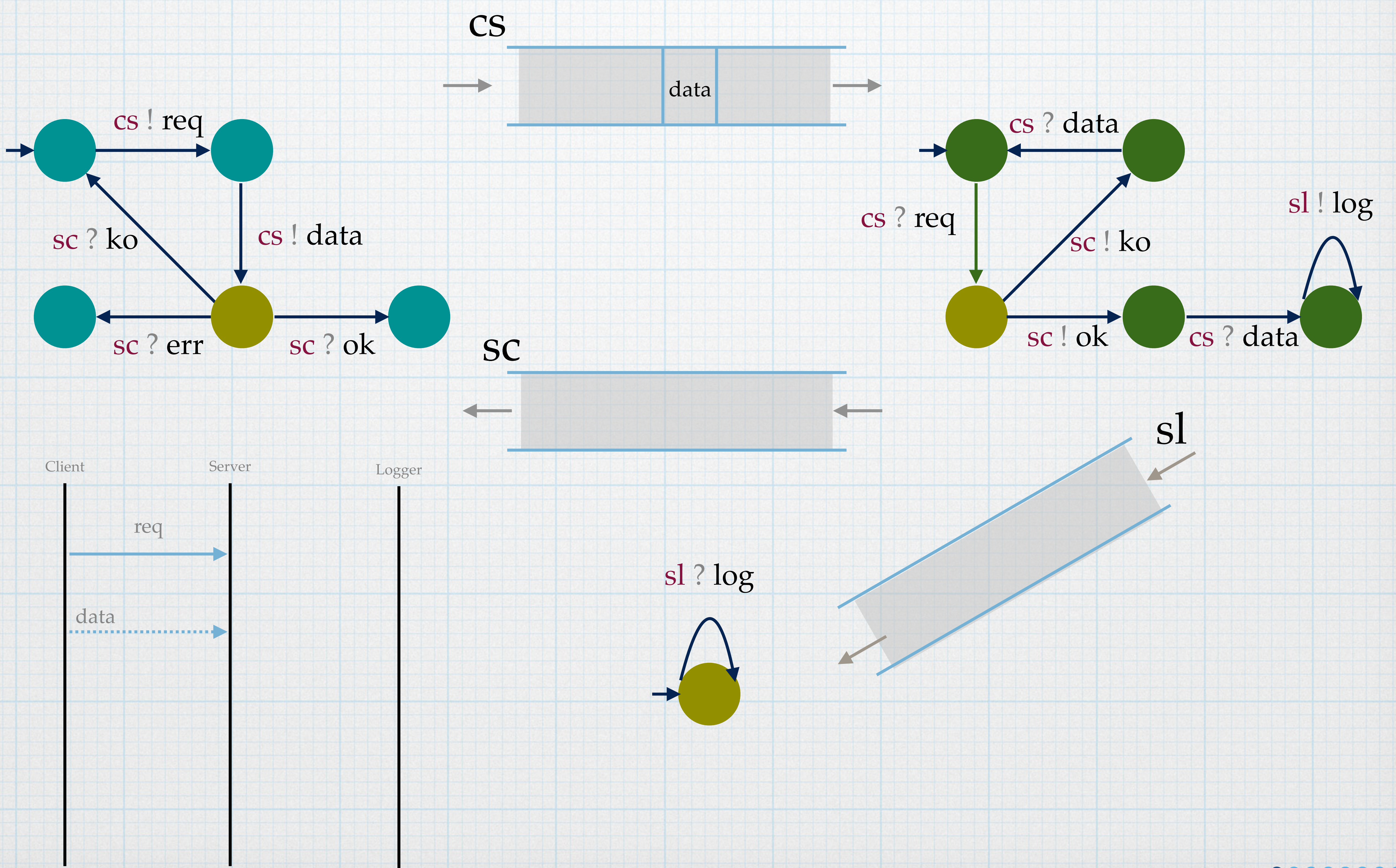
cs
req
cs ! req
cs ! data
sc ? ko
sc ? err
sc ? ok
cs ? data
cs ? req
sc ! ko
sl ! log
sc ! ok
cs ? data
sc
sl
Client
Server
Logger
req
sl ? log

The model



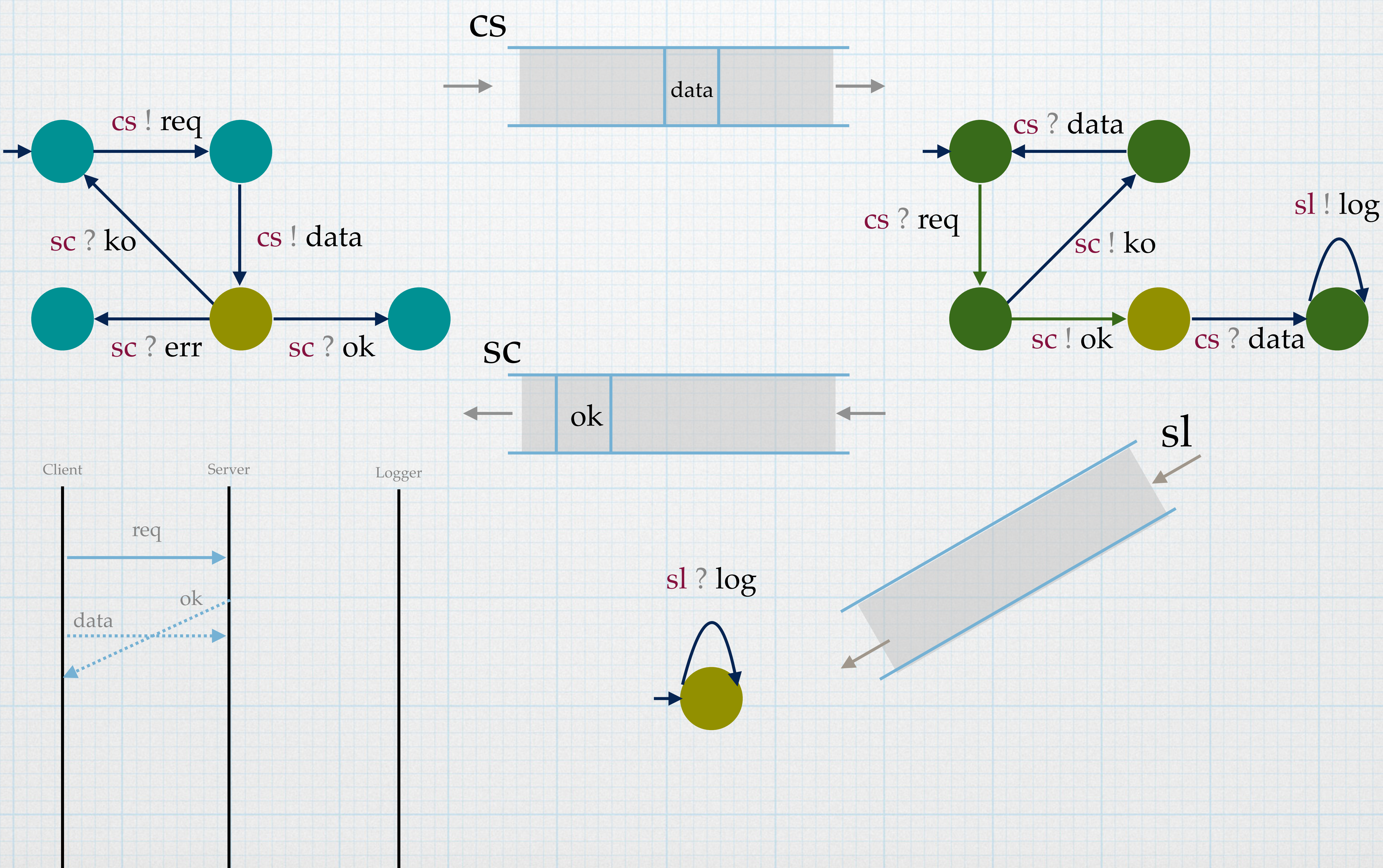
cs
data
req
cs ! req
sc ? ko
cs ! data
sc ? err
sc ? ok
cs ? data
cs ? req
sc ! ko
sl ! log
sc ! ok
cs ? data
sc
sl
Client
Server
Logger
req
data
sl ? log

The model

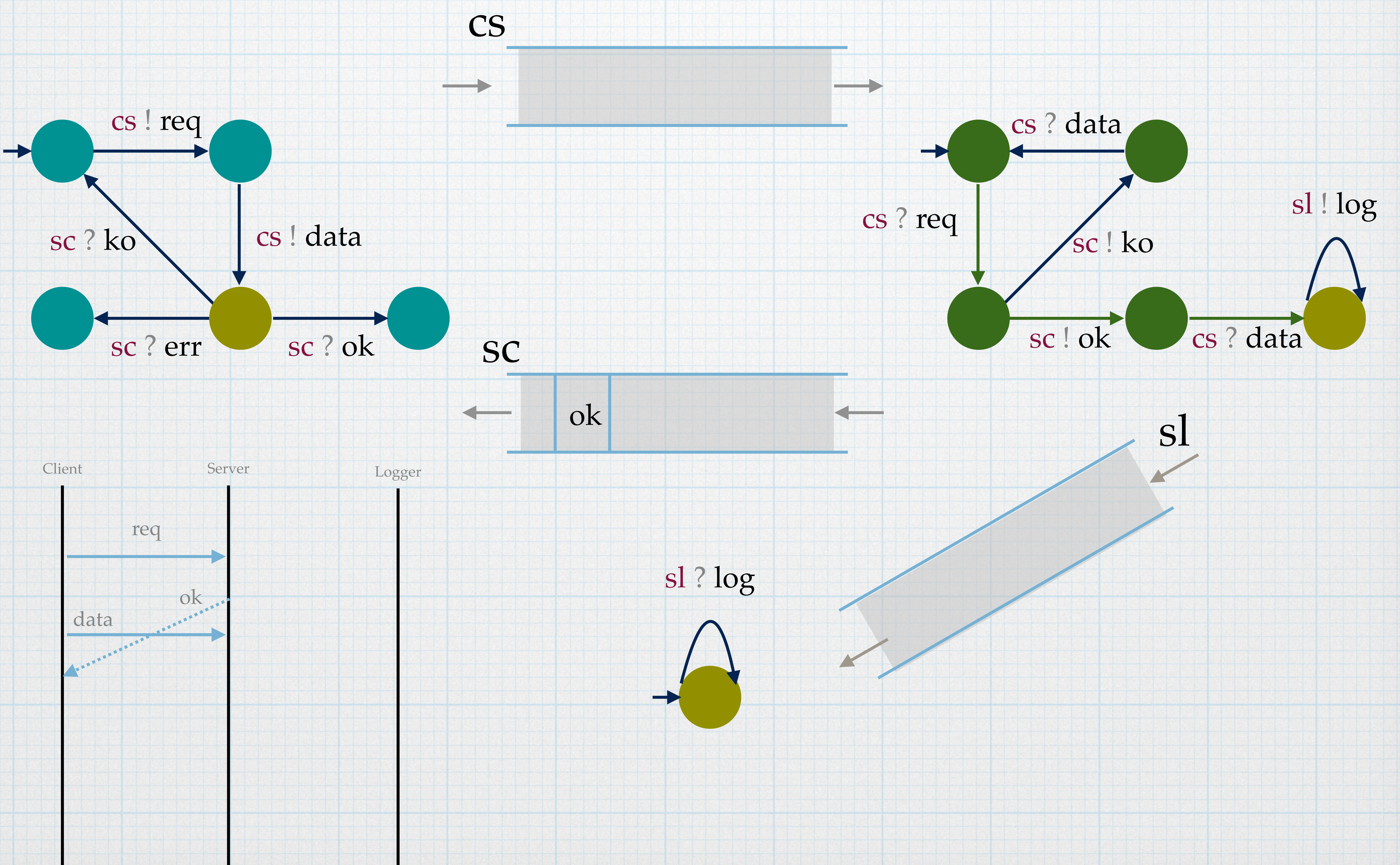


cs
data
cs ! req
cs ! data
sc ? ko
sc ? err
sc ? ok
sc
cs ? data
cs ? req
sc ! ko
sl ! log
sc ! ok
cs ? data
sl
Client
Server
Logger
req
data
sl ? log

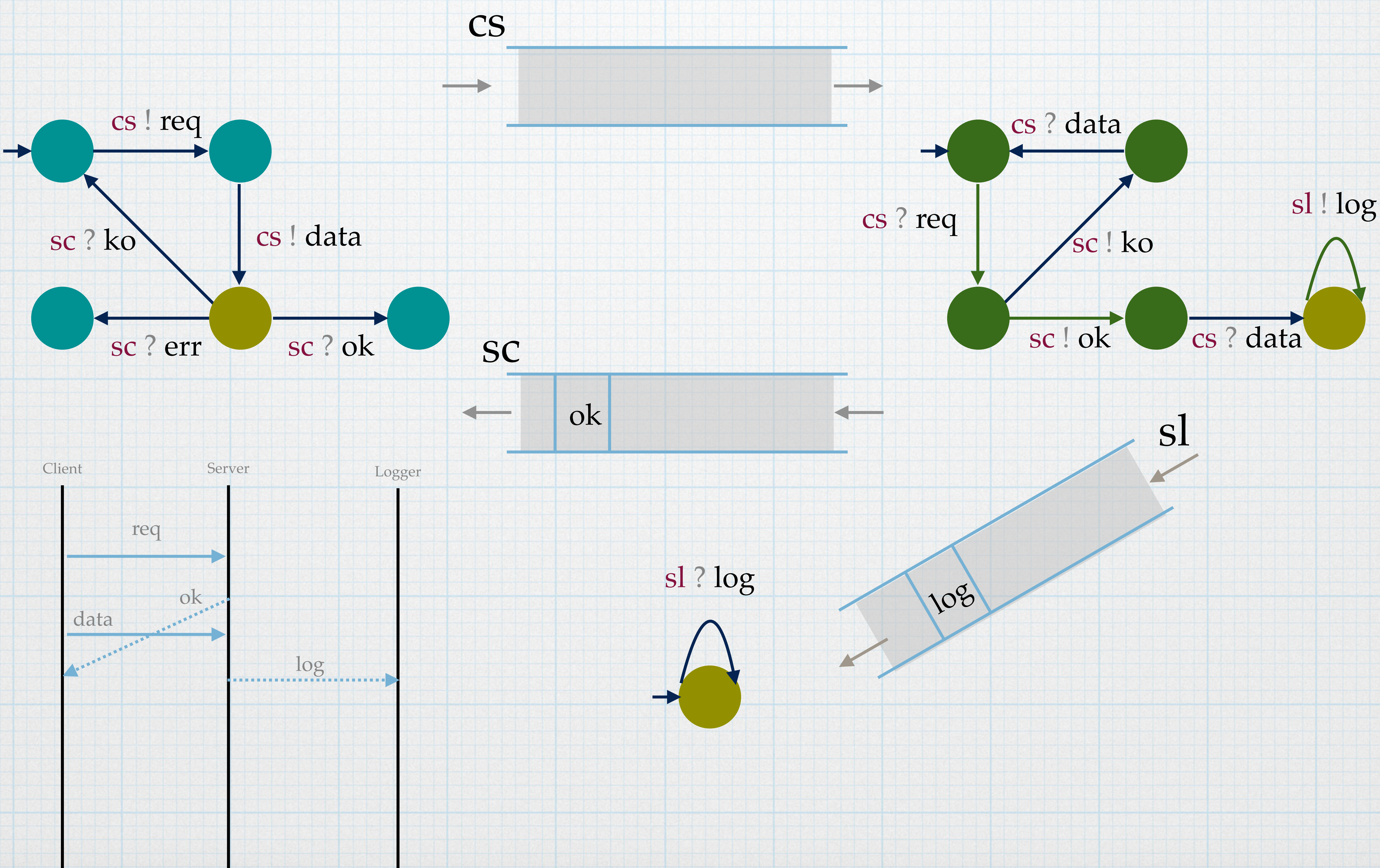
The model


cs
data
cs ! req
cs ! data
sc ? ko
sc ? err
sc ? ok
cs ? data
cs ? req
sc ! ko
sl ! log
sc ! ok
cs ? data
sc
ok
sl
Client
Server
Logger
req
ok
data
sl ? log

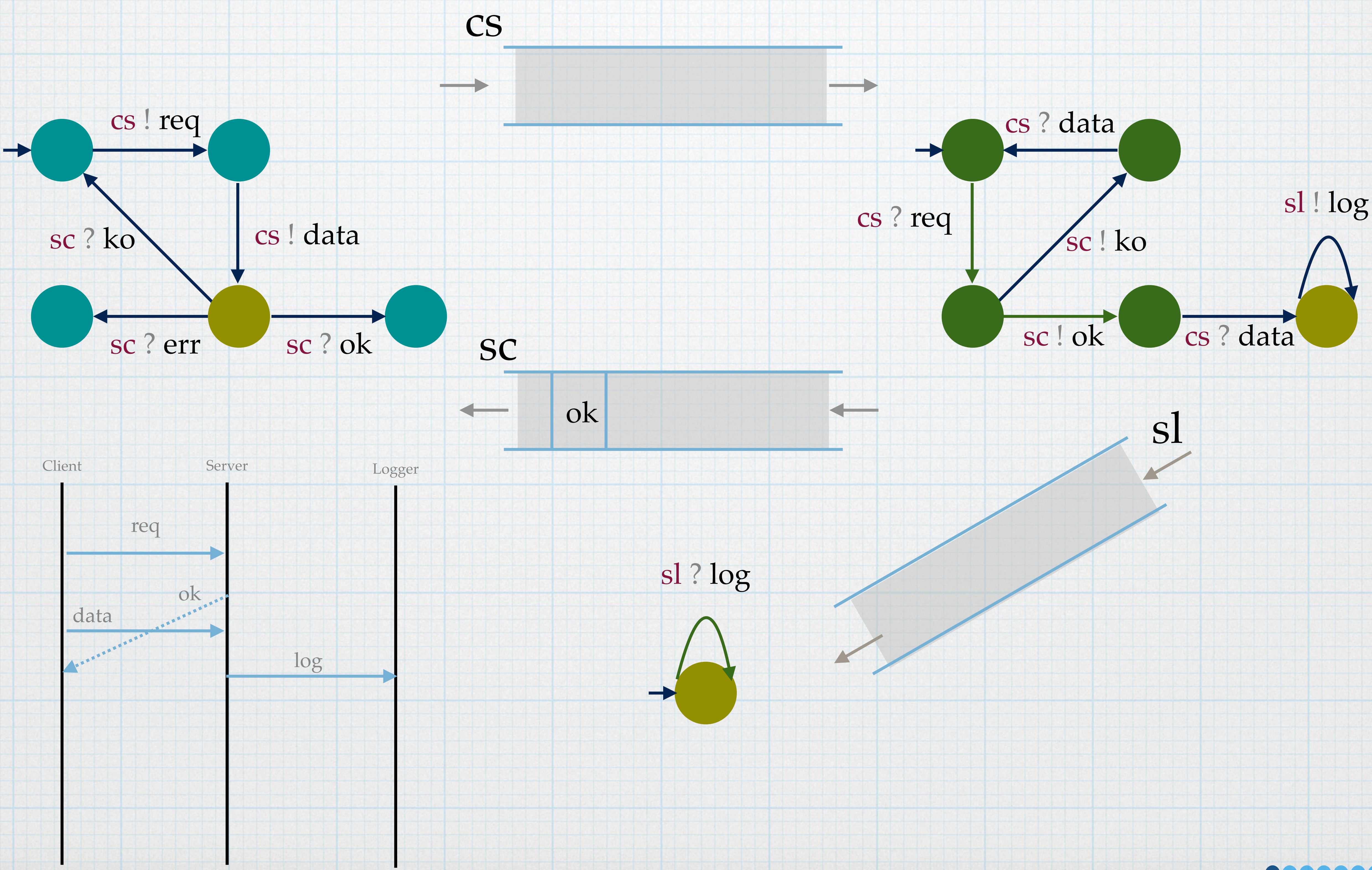
The model


cs
cs ! req
cs ! data
sc ? ko
sc ? err
sc ? ok
cs ? data
cs ? req
sc ! ko
sl ! log
sc ! ok
cs ? data
sc
ok
sl
Client
Server
Logger
req
ok
data
sl ? log

The model

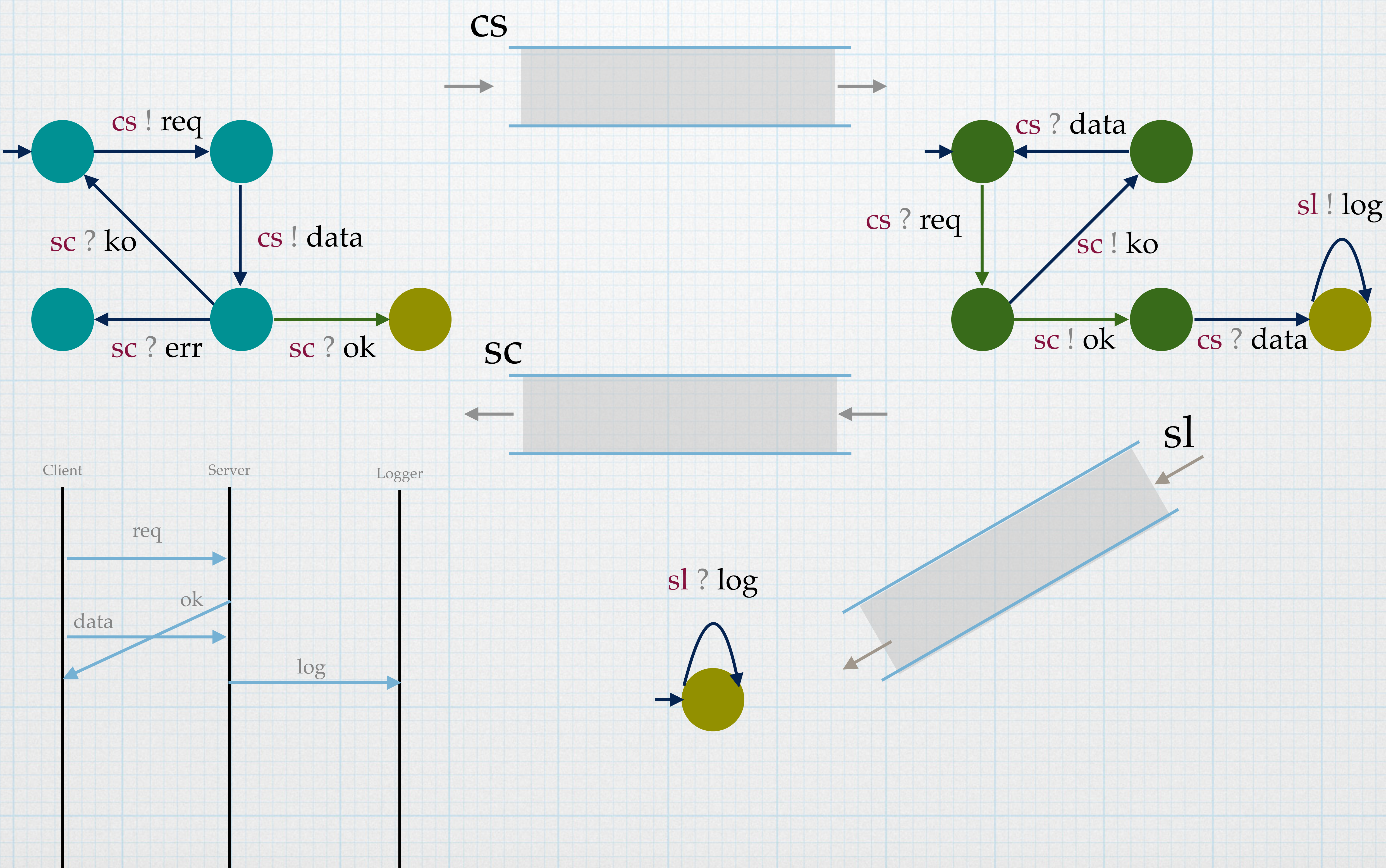

cs
cs ! req
sc ? ko
cs ! data
sc ? err
sc ? ok
cs ? data
cs ? req
sc ! ko
sl ! log
sc ! ok
cs ? data
sc
ok
sl
Client
Server
Logger
req
ok
data
log
sl ? log
log

The model



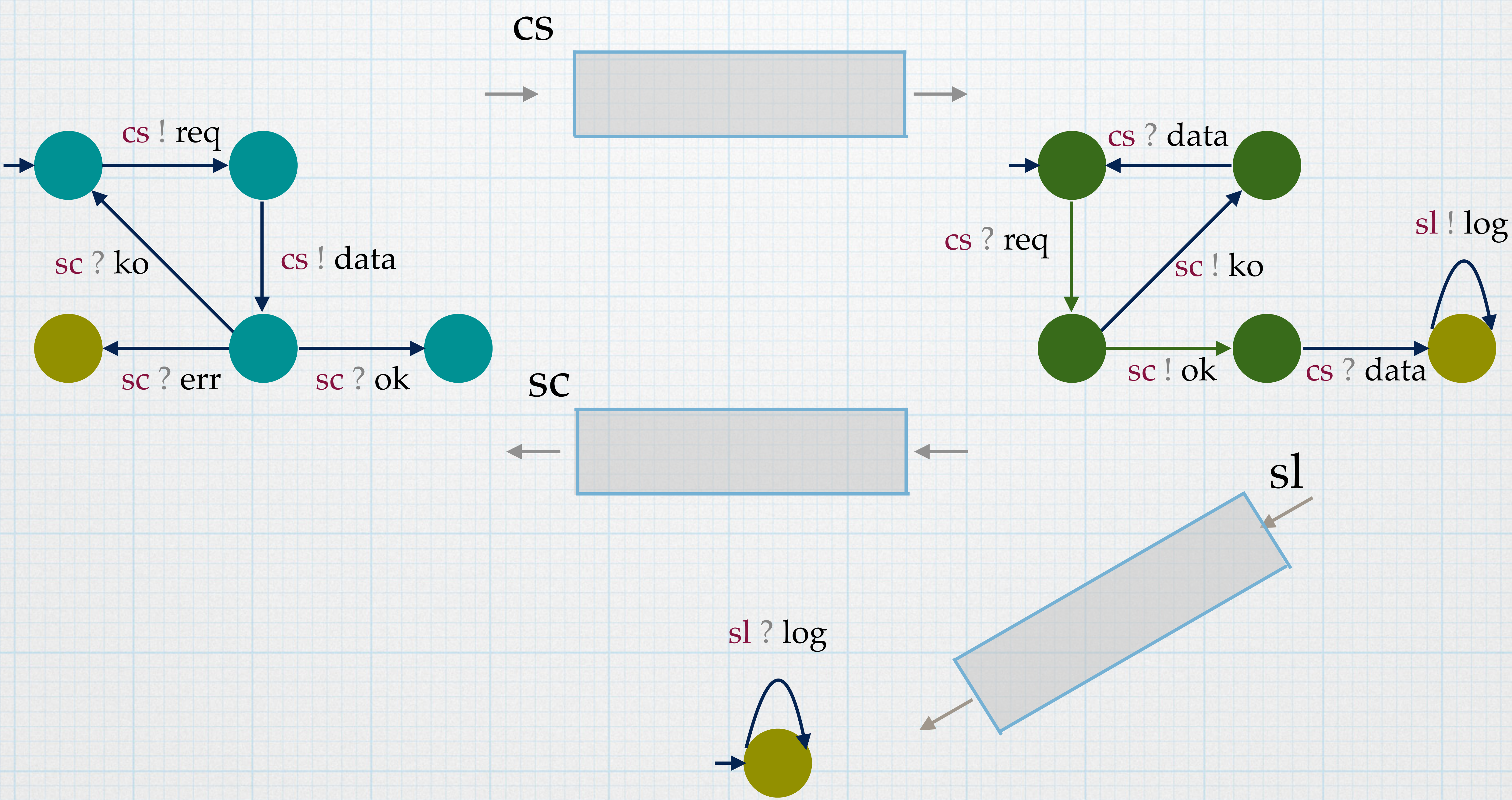
cs
cs ! req
cs ! data
sc ? ko
sc ? err
sc ? ok
cs ? data
cs ? req
sc ! ko
sl ! log
sc ! ok
cs ? data
sc
ok
sl
Client
Server
Logger
req
ok
data
log
sl ? log

The model



cs
sc
sl
cs ! req
cs ! data
sc ? ko
sc ? err
sc ? ok
cs ? data
cs ? req
sc ! ko
sl ! log
sc ! ok
cs ? data
sl ? log
Client
Server
Logger
req
ok
data
log

The Boundedness Problem



Does the model need an infinitely long channel?

The Boundedness Problem

* Turing-hard for communicating systems [BZ83]

* Therefore, there is a need for under-approximations

Bounded language

* A language is bounded if it is of the form:

  $u^* v^* w^* \ldots$ for words u, v, w…

* e.g. $(ab)^* (cd)^* a^*$ is bounded but

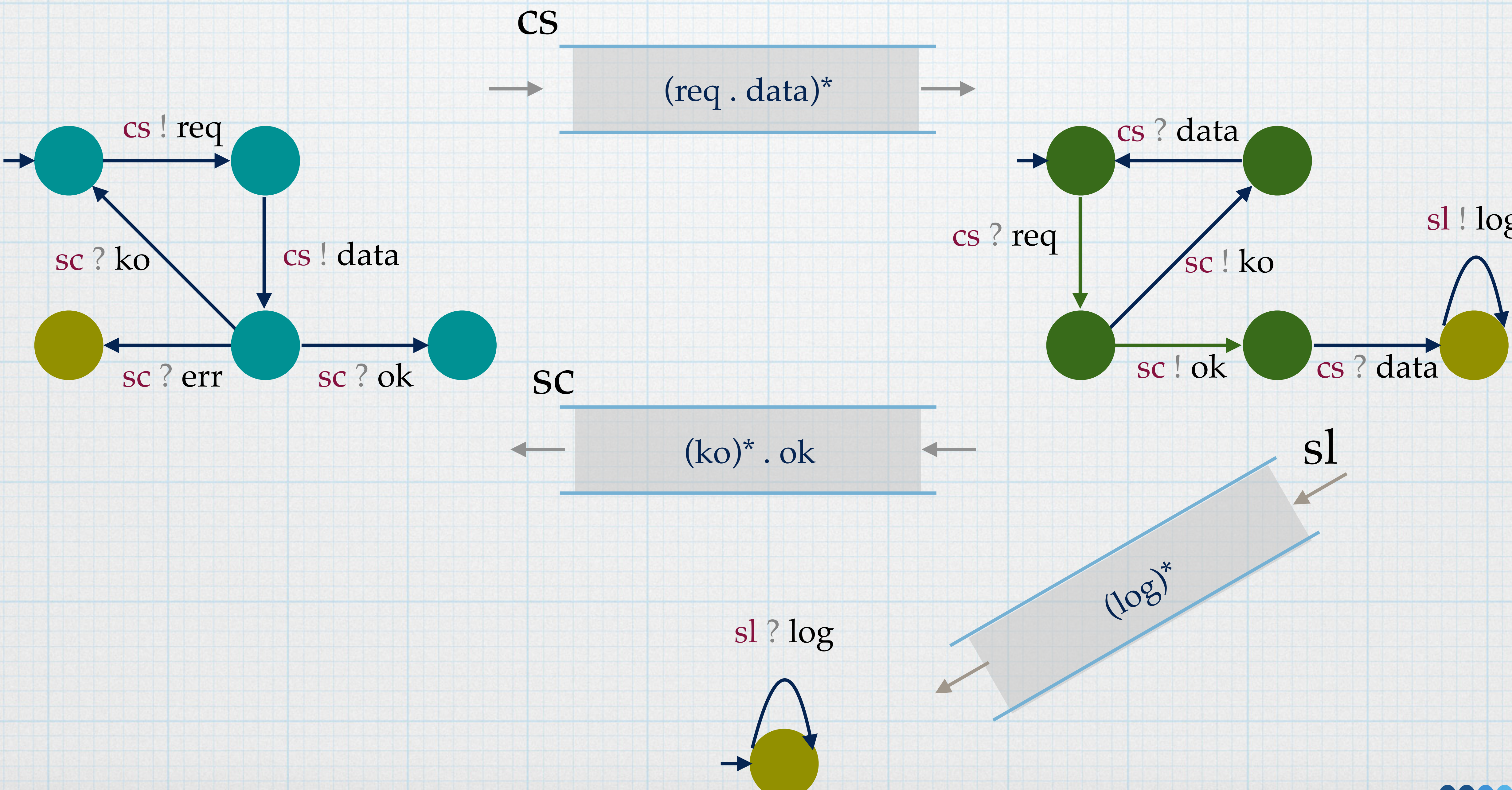
  $(a (ab)^* c)^*$ is not

Input-bounded channel

* If the language that enters the channel is bounded

Input-bounded channel

* If the language that enters the channel is bounded



Result I

[BFS20, BFS22a]

* Reachability (and many other verification problems) are decidable for input-bounded systems (where are channels are input-bounded)

Synchronizability

* A **synchronous** run:

  one where the channel size can be zero

Synchronizability

* A synchronizable run:

  every run is *equivalent* to a synchronous run

Result II

* Framework for deciding synchronizability for a variety of definitions [BGF+21]

Other under-approximations

* Branch-WSTS [BFS22b]

  - a large class of systems for which we can decide boundedness and termination

Other under-approximations

* Send-synchronizable systems

  - the receptions can be reordered to become synchronous

Other under-approximations

* Reversal-bounded systems

  - limit the number of alternations of sending and reception

FUTURE WORK

Half-Duplex systems

[CF05]

* For every pair of processes, at most one of the channels between them is non-empty

Half-Duplex systems

Half-duplex

Not half-duplex

Half-Duplex systems

* For two processes, membership (and some other verification problems are decidable.
* For three processes, it is Turing-hard.

Application in the realm of channel contracts

* Half-duplex contracts can be seen as a way of defining reliable contracts [LV11]
* We obtain determinism and uniform choice
* Robust in the presence of error-prone communication

Open question I

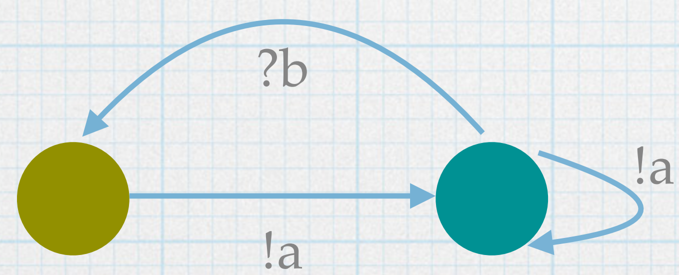
* Can we modify the definition for more than 2 processes? [DGGL21]

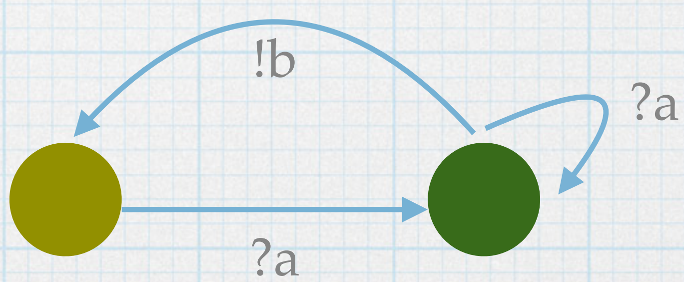
* What is class of “reliable” channel contracts for multiparty FIFO systems?

[LY19]

Communicating Session Automata

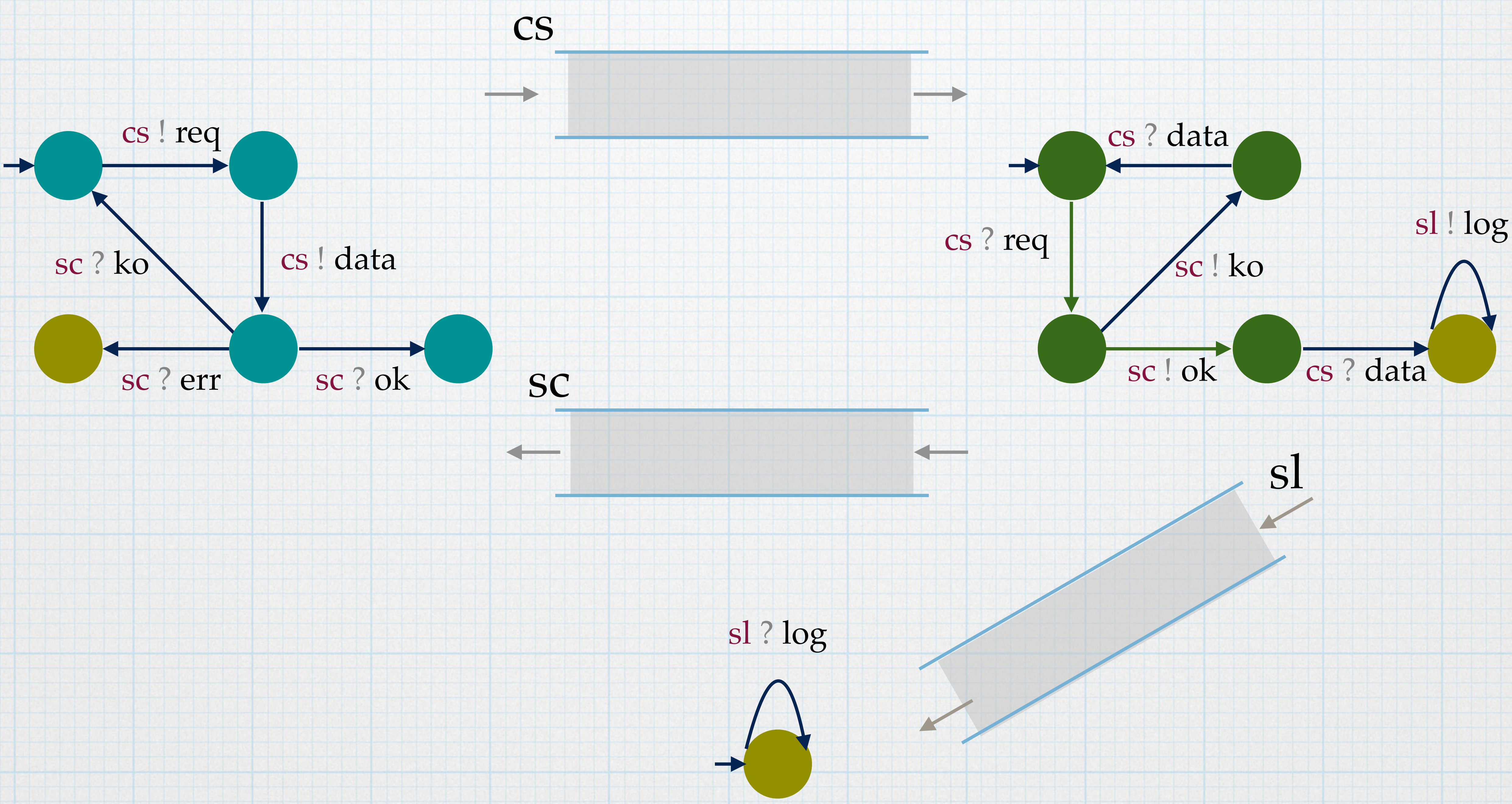
* Deterministic automata and each control state is uniquely sending or receiving





Not a communicating session automata

Communicating Session Automata



Multiparty compatibility

* Absence of deadlocks, orphan messages, unspecified receptions
* Whenever a message can be sent, it should eventually be sent

Multiparty compatibility

* Captures asynchrony
* Sufficient condition of existential boundedness
* Each automaton behaves similarly when you take any larger bound

Open question II

* Does the unifying framework capture this property?
* Can we try to extend this class to include mixed states?

Mid-term objectives

* A practical tool for verifying synchronizability (and other under-approximations of communicating automata) in the spirit of earlier works [LY19, BEJQ18]

Mid-term objectives

* Studying the relationship between k-synchronizability and high-level message sequence charts (HMSCs)
* Is an HMSC computable from a k-synchronous system?
* Can we characterize them?

Long-term objectives

* Adapting existing results to build a bounded model-checking strategy for general CFSMs (and the same for input-bounded)
* Using session types to formally model choreography languages and characterize realizability
* Extending the notions of synchronizability and choreography realizability for other communication models (bags, causally ordered channels, etc.)

References

* [BEJQ18] Ahmed Bouajjani, Constantin Enea, Kailiang Ji, and Shaz Qadeer. *On the completeness of verifying message passing programs under bounded asynchrony*, CAV 2018.
* [BFG+21] Benedikt Bollig, Cinzia Di Giusto, Alain Finkel, Laetitia Laversa, Etienne Lozes, and Amrita Suresh. *A Unifying Framework for Deciding Synchronizability*, CONCUR 2021.
* [BFS20] Benedikt Bollig, Alain Finkel, and Amrita Suresh. B*ounded Reachability Problems Are Decidable in FIFO Machines*, CONCUR 2020.
* [BFS22a] Benedikt Bollig, Alain Finkel, and Amrita Suresh. *Bounded Problems Are Decidable in FIFO Machines*, LMCS 2022.
* [BFS22b] Benedikt Bollig, Alain Finkel, and Amrita Suresh. *Branch-Well-Structured Transition Systems and Extensions*, FORTE 2022.

References

* [BZ83] Daniel Brand and Pitro Zafiropulo. *On communicating finite-state machines*, J. ACM 1983.
* [CF05] Gerald Cécé and Alain Finkel. *Verification of Programs with Half-Duplex Communication*, Inf Comp 2005.
* [DGGL21] Cinzia Di Giusto, Loïc Germerie Guizouarn, and Etienne Lozes. *Multiparty Half-Duplex Systems and Synchronous Communications*, ICE 2021.
* [LV11] Étienne Lozes and Jules Villard. *Reliable Contracts for Unreliable Half-Duplex Communications*, WS-FM 2011.
* [LY19] Julien Lange and Nobuko Yoshida. *Verifying Asynchronous Interactions via Communicating Session Automata*, CAV 2019.