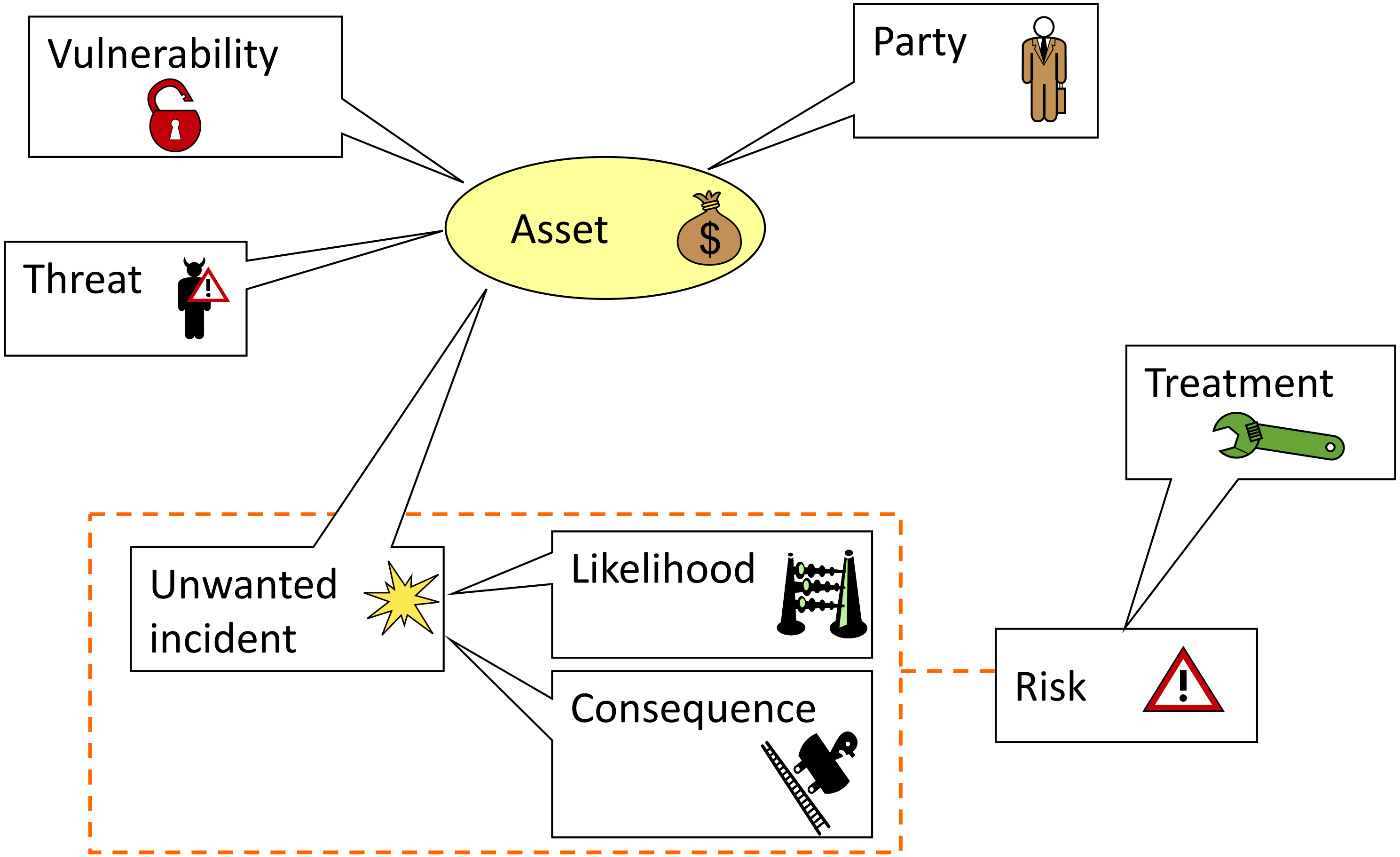
1a) How should the concepts in this drawing below be represented in a sequence diagram?

1b) Use this representation to model a DDOS attack.



# 1a) How should the concepts in this drawing below be represented in a sequence diagram?

Asset – lifeline or attribute

Party – normally not in sequence diagram

Threat – lifeline

Unwanted incident – event

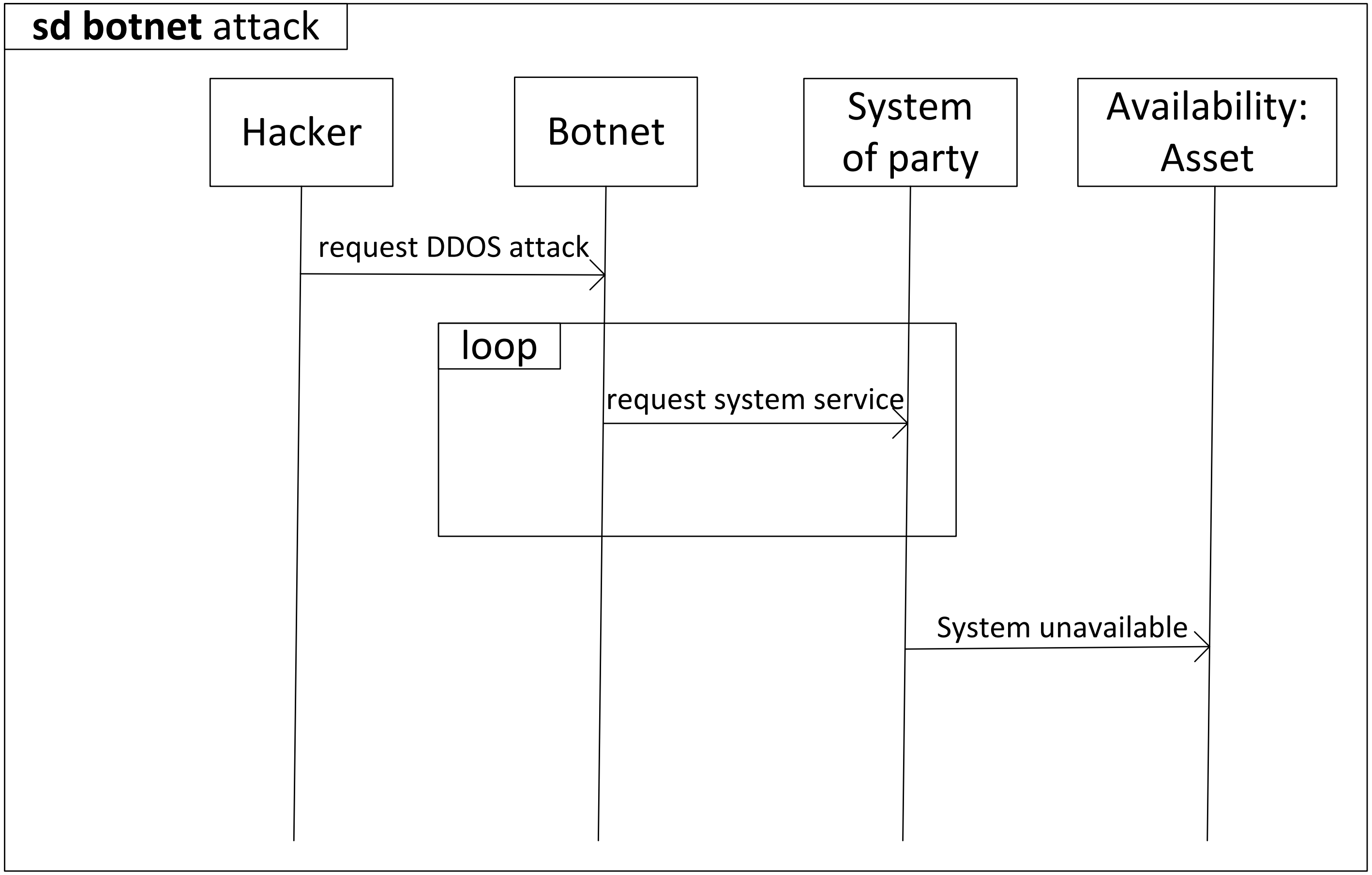
Likelihood – annotation on message arrow

Consequence – annotation on message arrow

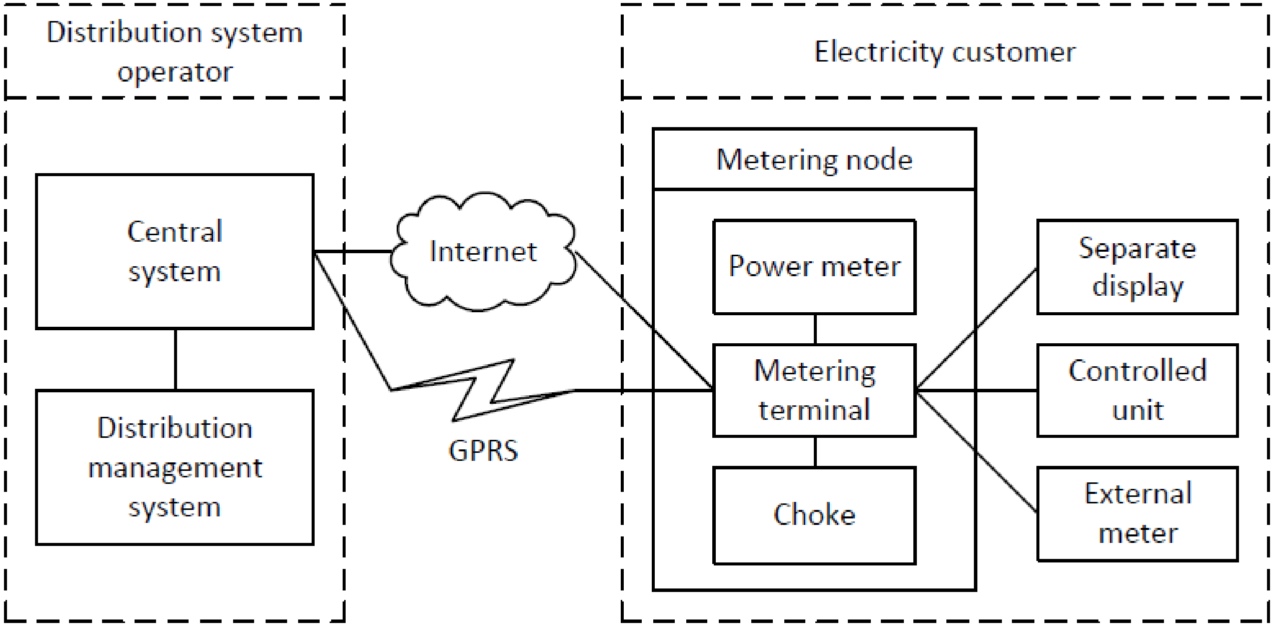
Vulnerability – case dependent

Treatment – case dependent

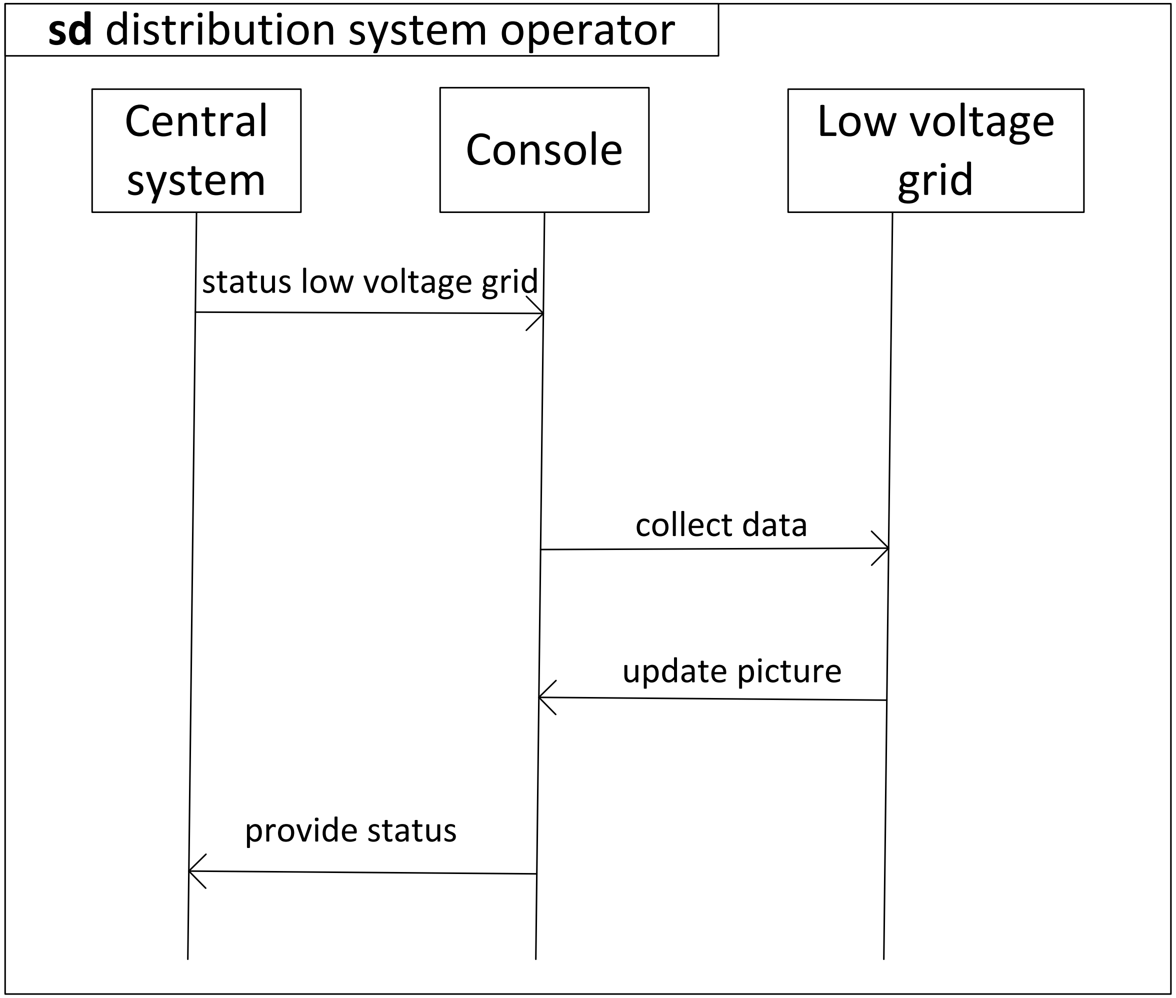
# 1b) model of a DDOS attack



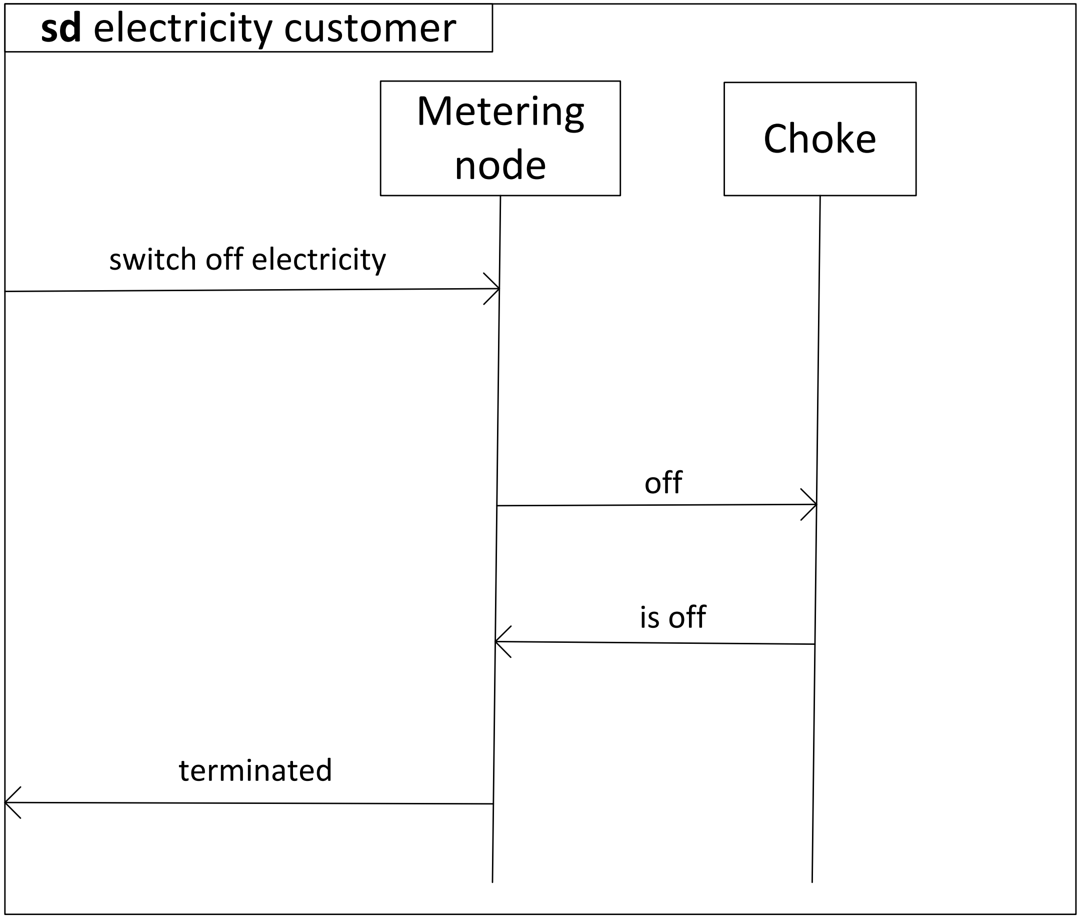
- 2a) Make a sequence diagram for the distribution system oprator;
- 2b) Do the same the electricity customer;
- 2c) Make a sequence diagram describing an external attack on an electricity customer via the distribution system operator.



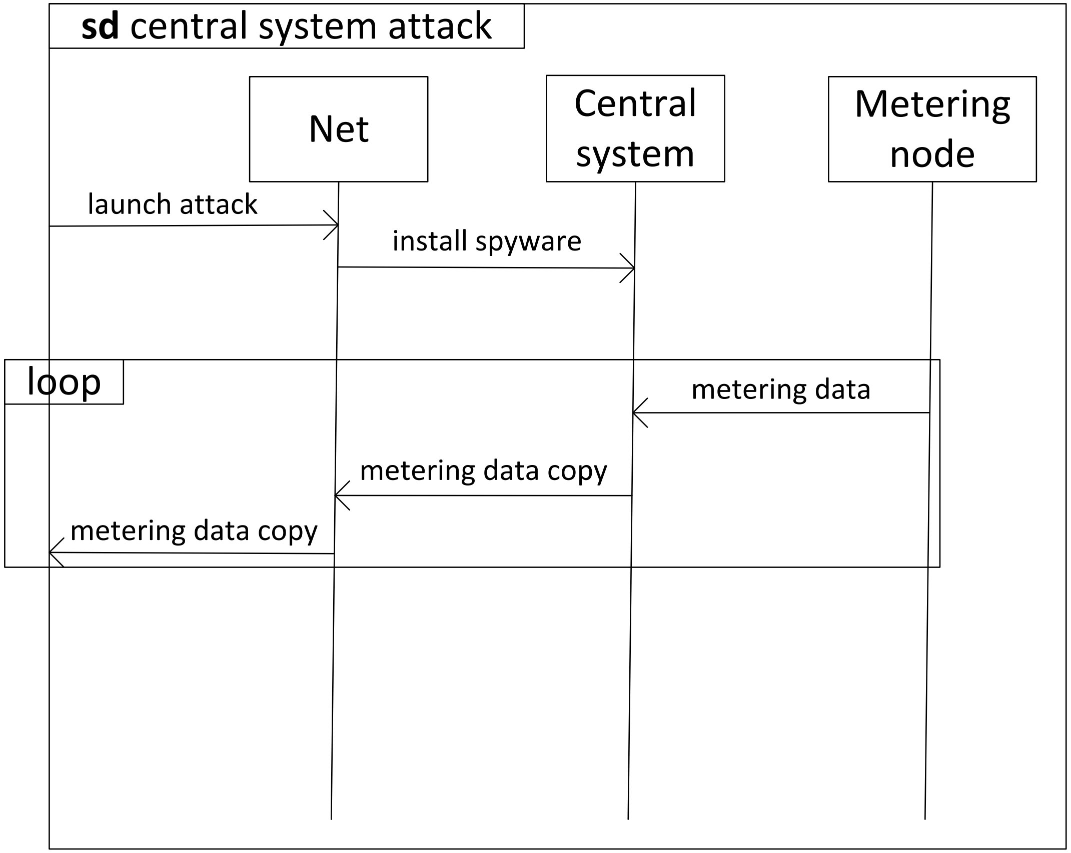
# 2a) sequence diagram for the distribution system oprator



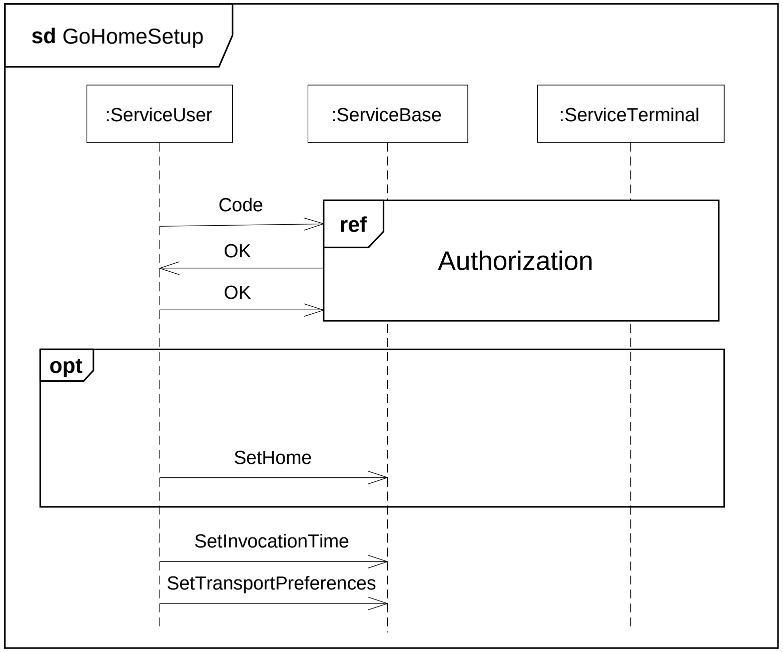
# 2b) sequence diagram for the electricity customer

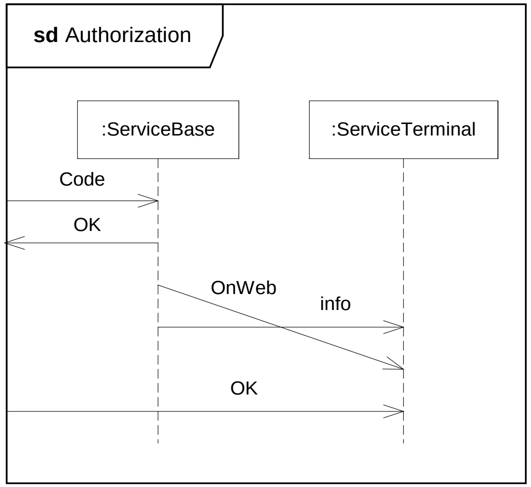


# 2c) sequence diagram for an external attack via the operator



a) Describe the traces of the opt-fragment;
b) Describe the traces of Authorization;
c) Describe the traces of GoHomeSetup (assuming that :ServiceBase and ServiceTerminal are merged into one lifeline)





# a) Describe the traces of the opt-fragment;

The **opt** operator means that we either do nothing (e.g. we **skip**) or we execute the body of the **opt**.

The result of skipping is the empty trace, which is written as follows: <>

The result of executing the body is: <!sh,?sh>, where "sh" is short for SetHome

Hence, we get two traces.

# b) Describe the traces of Authorization;

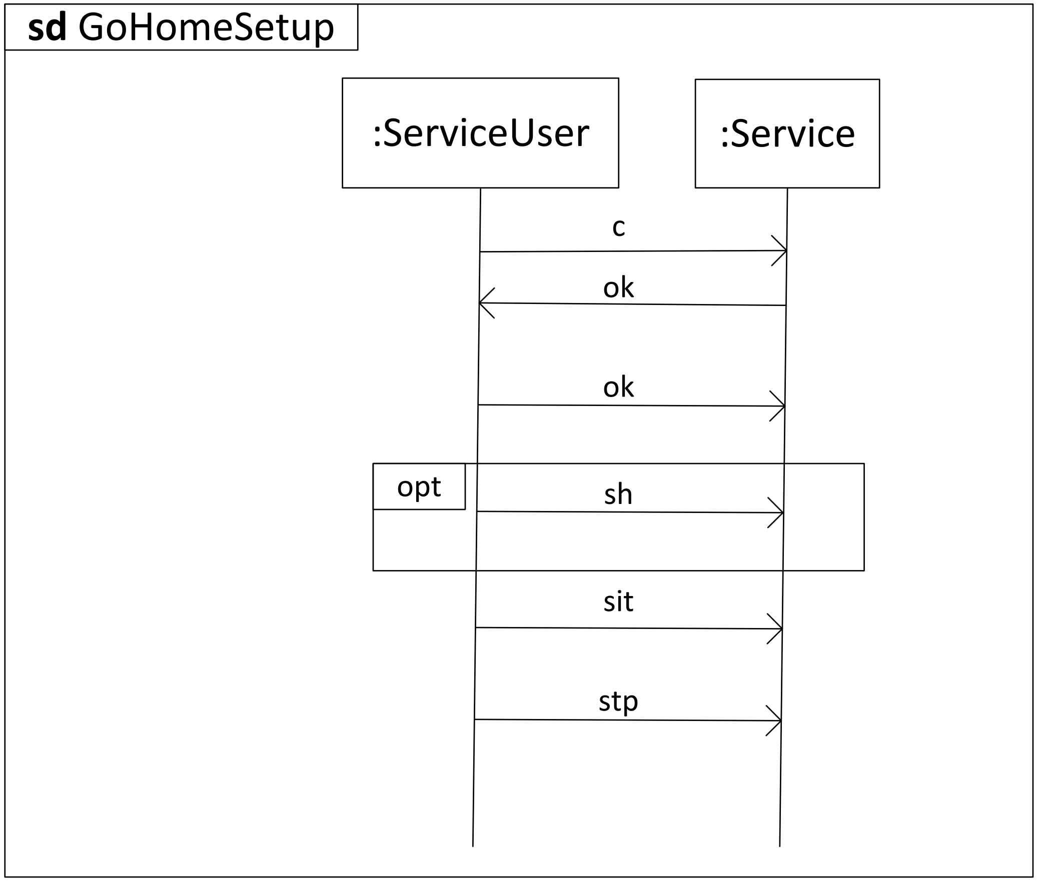
Remember that a trace consists of events only.

Events takes place only at lifelines.

This means that Authorization has only one trace:

<?Code,!Ok,!OnWeb,!Info,?Info,?OnWeb,?ok>

# c) Result of merging lifelines and simplifying message names



# c) Describe the traces of GoHomeSetup (assuming that :ServiceBase and :ServiceTerminal are merged into one lifeline)

Alle the traces start with following 5 events

<!c,?c,!ok,?ok,!ok, ….

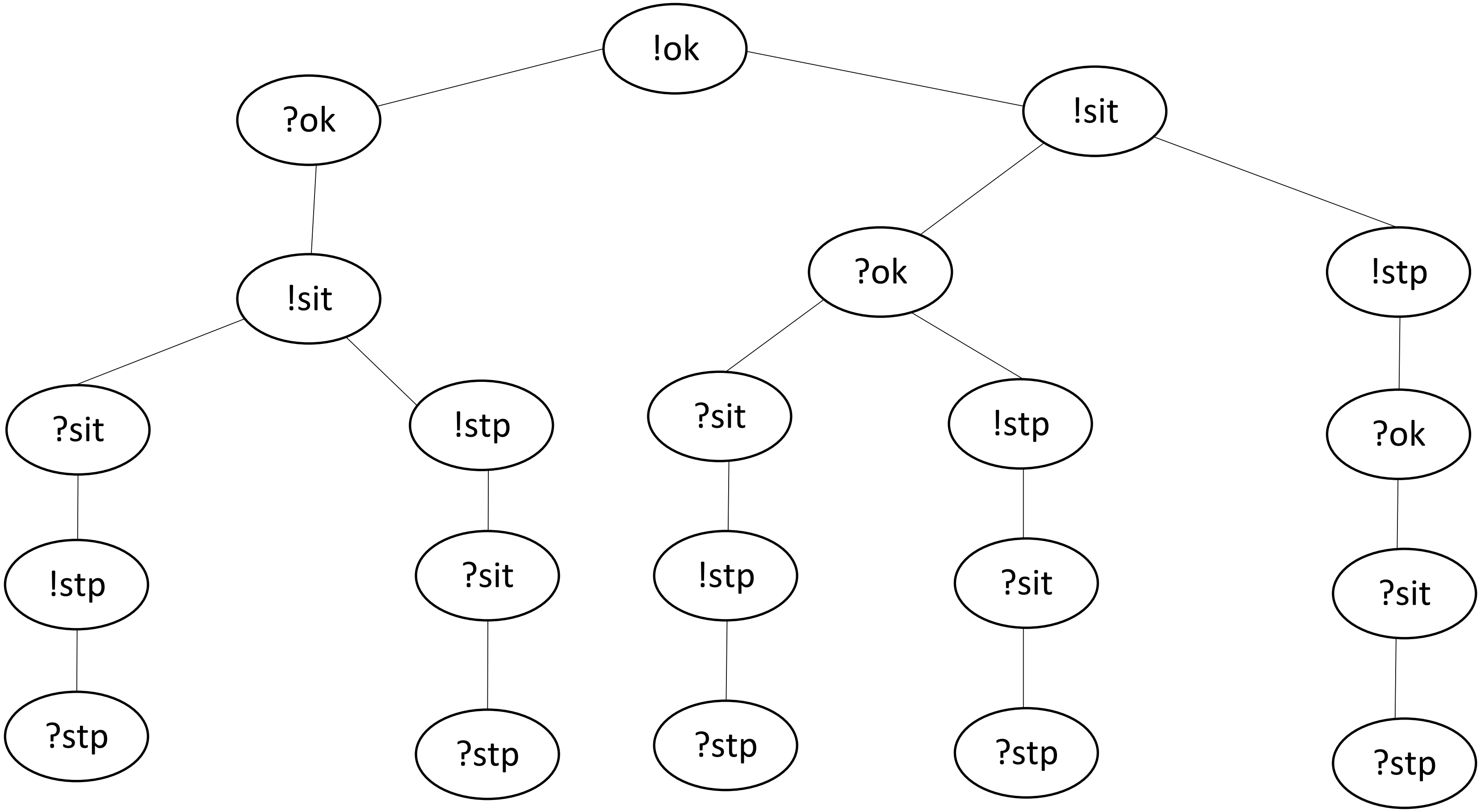
The trace trees on the next two pages therefore start from the last of these five events.

We distinguish between two cases:

Case A: the body of **opt** is skipped

Case B: the body of **opt** is executed

# Case A



!ok

?ok

!sit

?ok

!stp

!sit

?sit

!stp

?sit

!stp

?ok

!stp

?sit

!stp

?sit

?sit

?stp

?stp

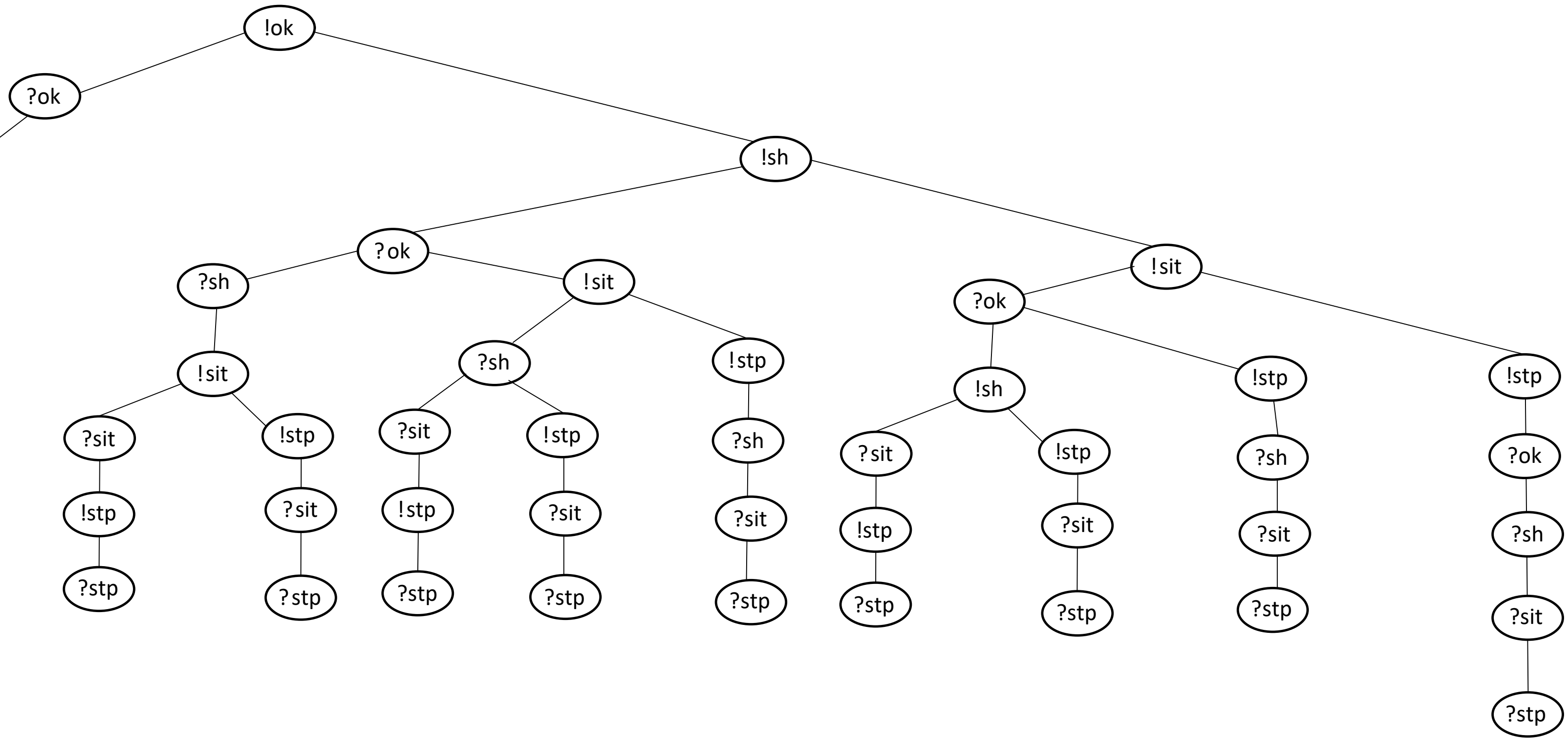
?stp

?stp

?stp

# Case B

Analogous with Case A: we get 5 branches; the only difference is that each ocurrence of «ok» is replaced by «sh».



!ok
?ok
!sh
?ok
!sit
?sh
!sit
?ok
!sit
?sh
!stp
!sh
!stp
!stp
?sit
!stp
?sit
!stp
?sh
?sit
!stp
?sh
?ok
!stp
?sit
!stp
?sit
?sit
!stp
?sit
?sit
?sh
?stp
?stp
?stp
?stp
?stp
?stp
?stp
?stp
?sit
?stp