POLITECNICO DI MILANO

Performance Evaluation and Applications





Introduction to Performance Modelling and Basic Measurements

Performance modeling

Performance Evaluation is the quantitative and qualitative study of systems, to evaluate, measure, predict and ensure target behaviors and performances.

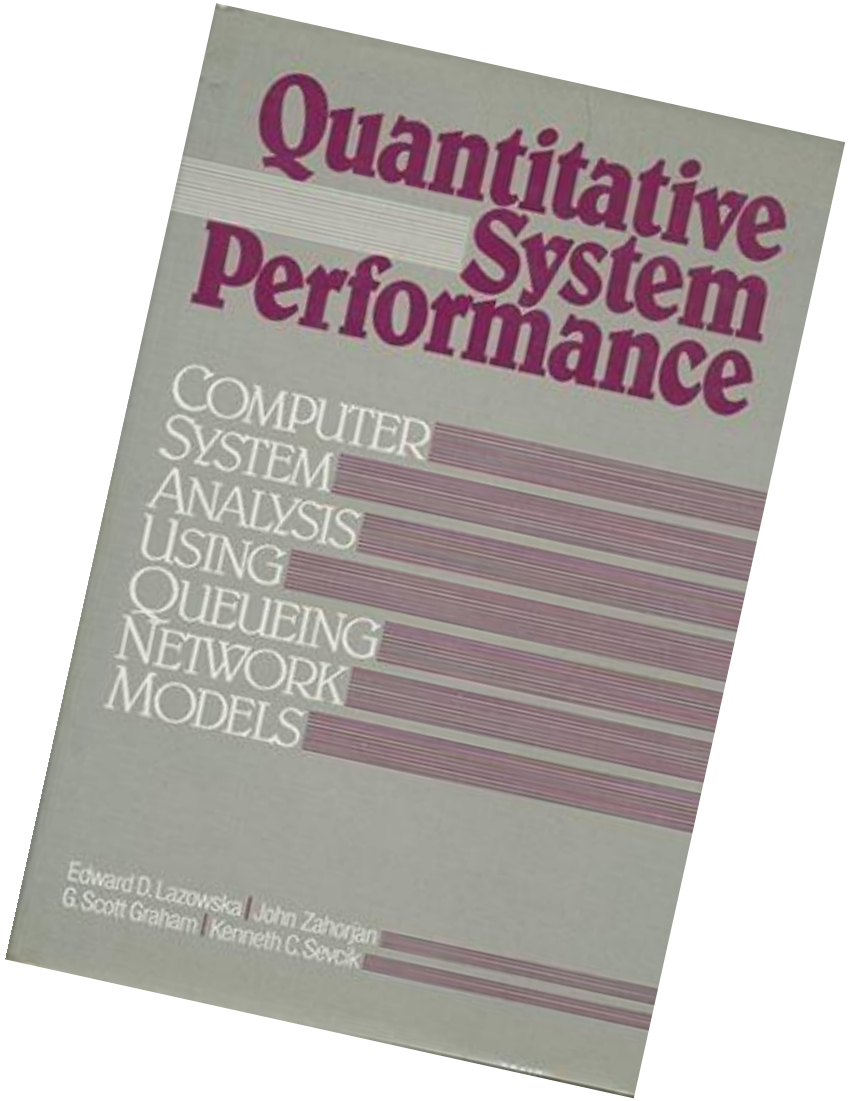
It is usually carried on using *models of a system*.

Performance modeling

A model is an abstraction of a system:

"an attempt to distill, from the details of the system, exactly those aspects that are essentials to the system behavior"....

(E. Lazoswka)

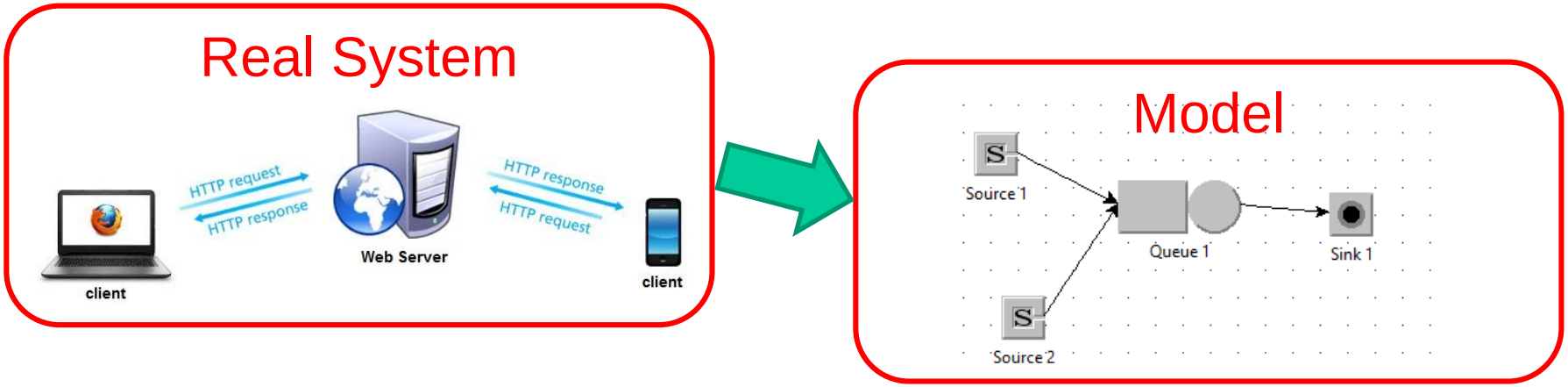


https://suno.com/song/b567d325-66a1-4c8c-94ce-1e8fe801e2cb

Models

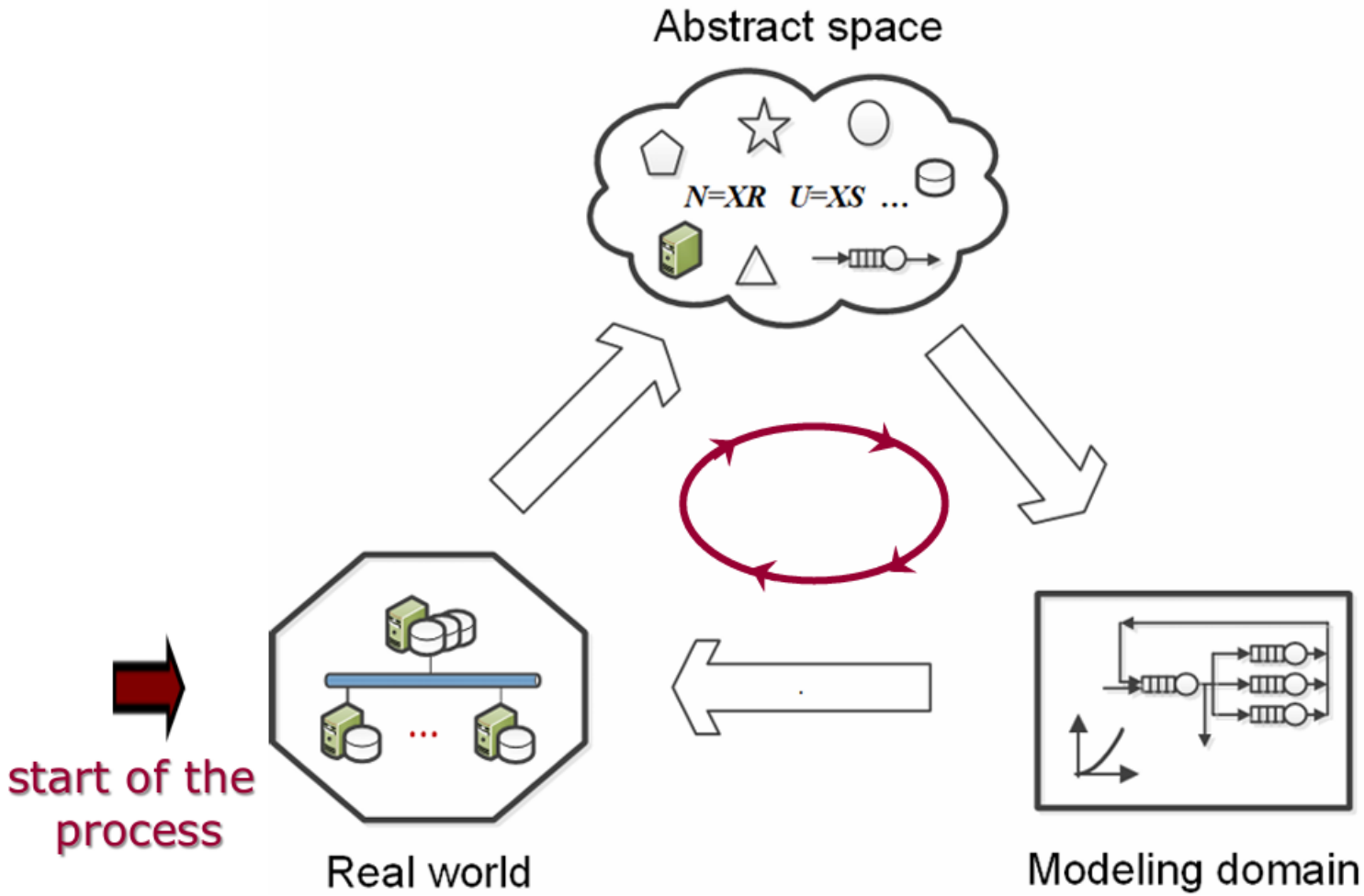
We abstract a system as a set of *Events and States* that describe the temporal evolution of some tasks.

The *model* defines which tasks are carried out, when they are executed, in which way they are selected to be run, how long they last, and many other details to closely match the real system. These details determines the events and the evolution of the state of the model.



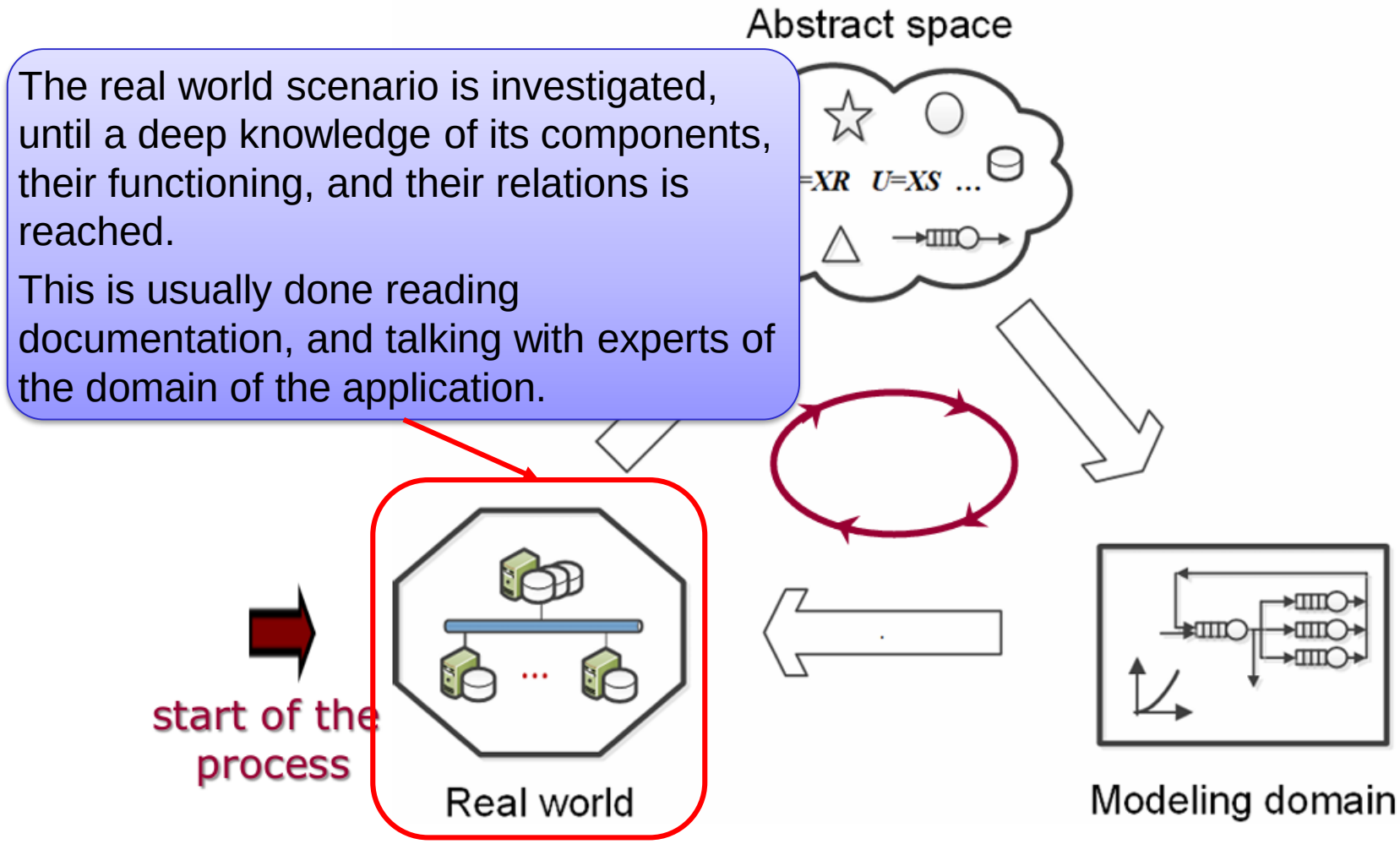
Environments involved in modeling

The modelling process requires many environments:



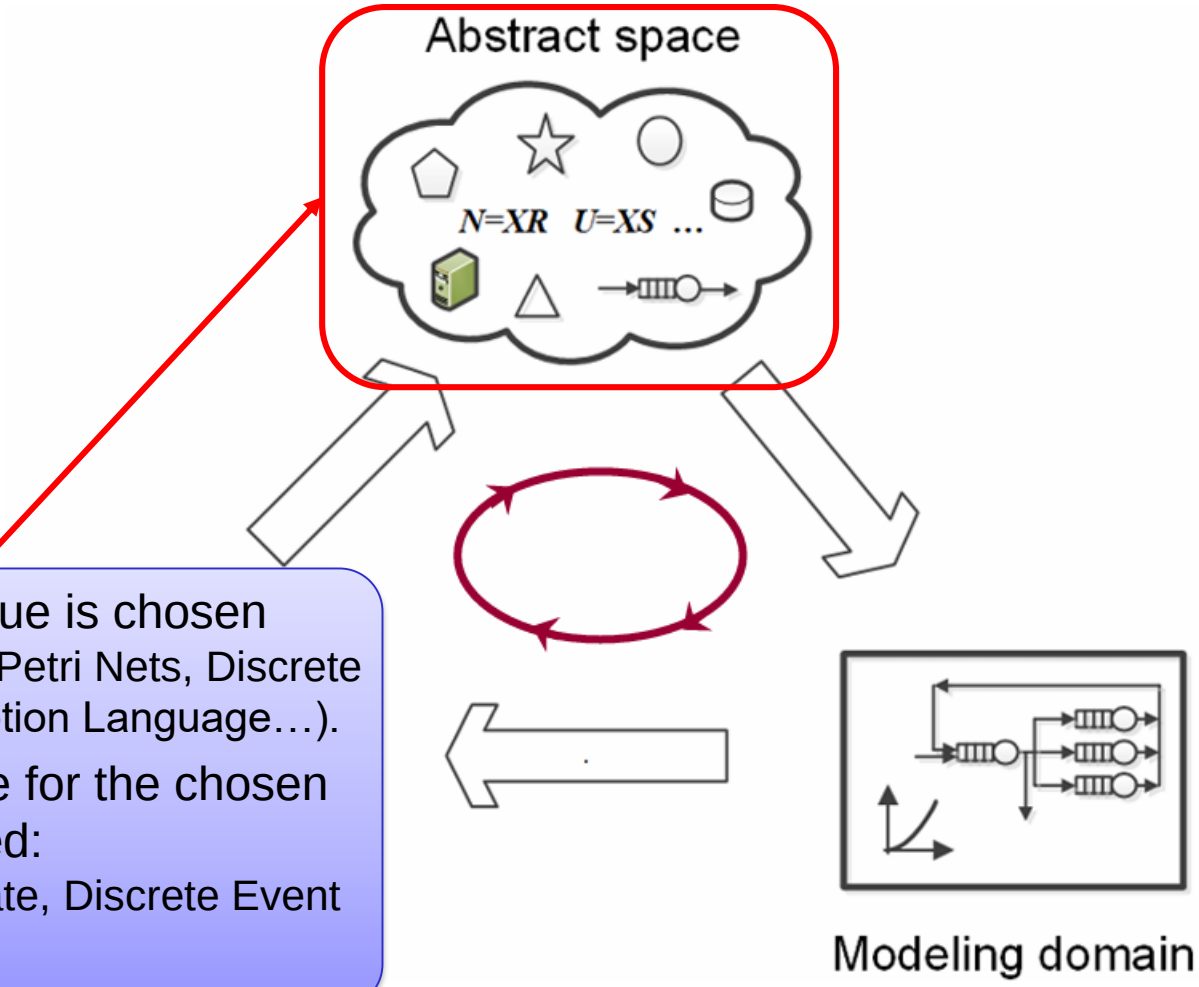
Environments involved in modeling

The modelling process requires many environments:



Environments involved in modeling

The modelling process requires many environments:

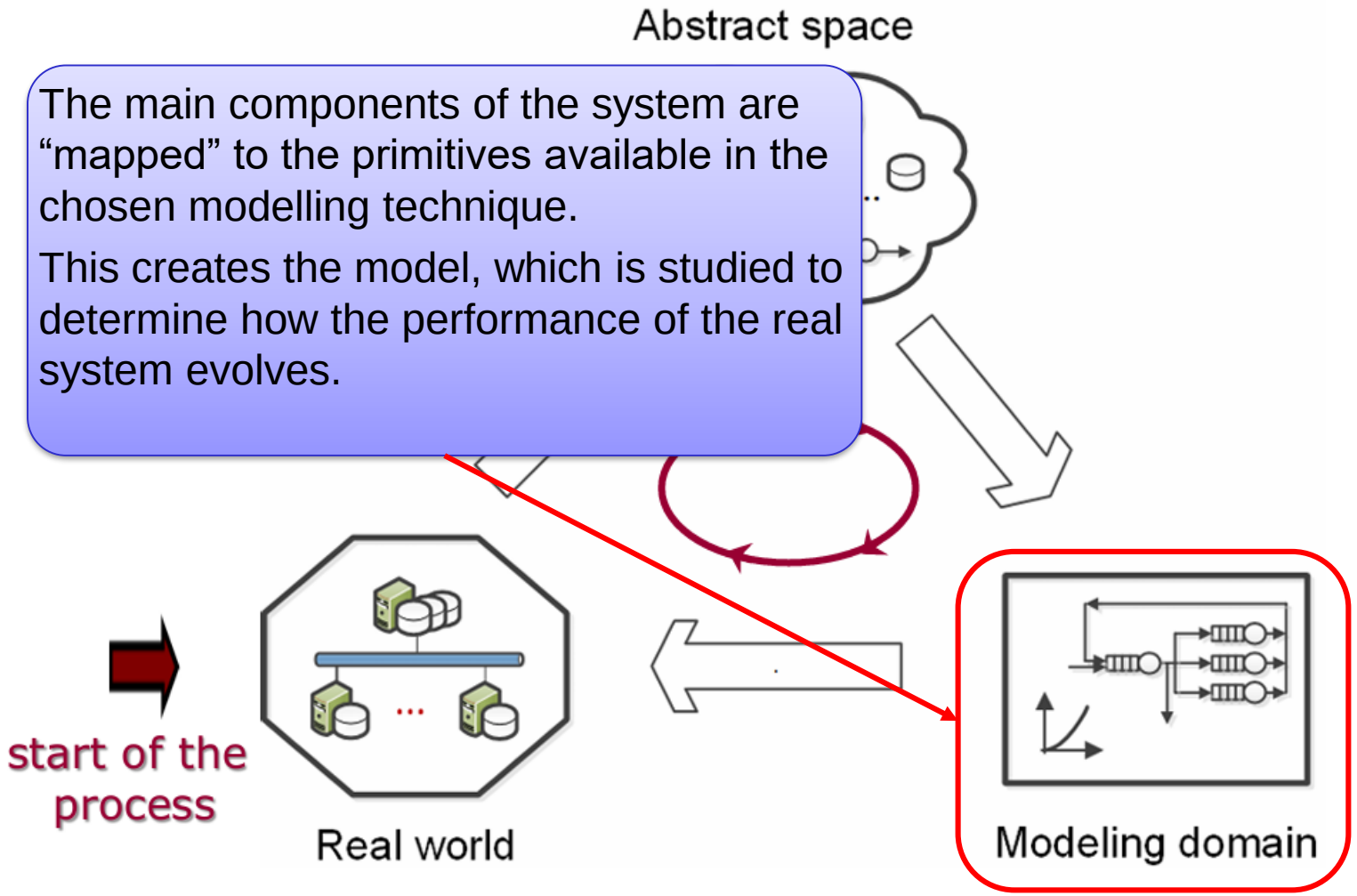


A modelling technique is chosen (Queueing Networks, Petri Nets, Discrete Event System Description Language…).

A solution technique for the chosen formalism is selected: (Analytical, Approximate, Discrete Event Simulation)

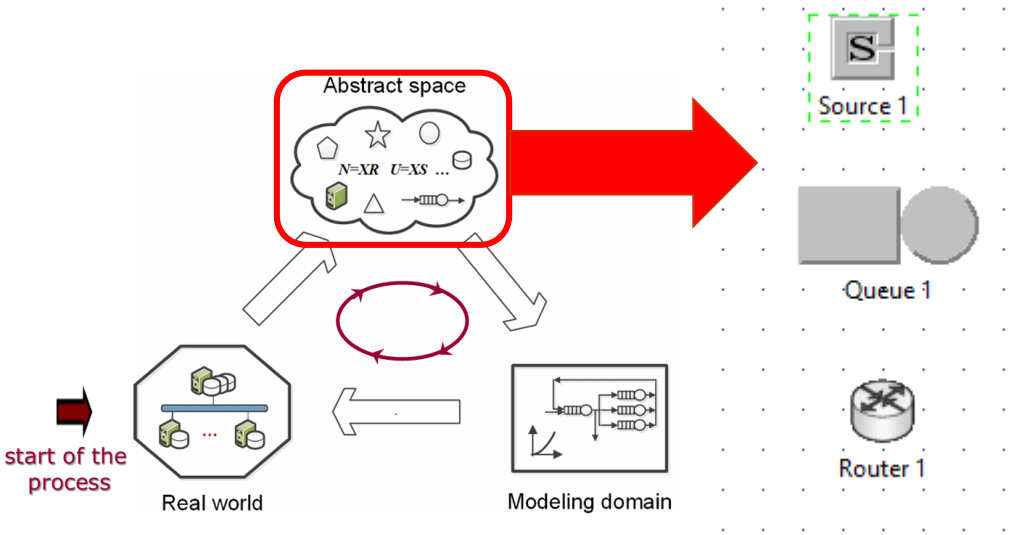
Environments involved in modeling

The modelling process requires many environments:



Environments involved in modeling

One of the main modelling environment we will focus on are the *Queueing Networks*, where a system is essentially modelled by a set of *Stations* that performs different *Task* to complete a *Job*. Once a task at a station is completed, it is *Routed* to other stations for further processing, or it is completed.



Sources of *Jobs*

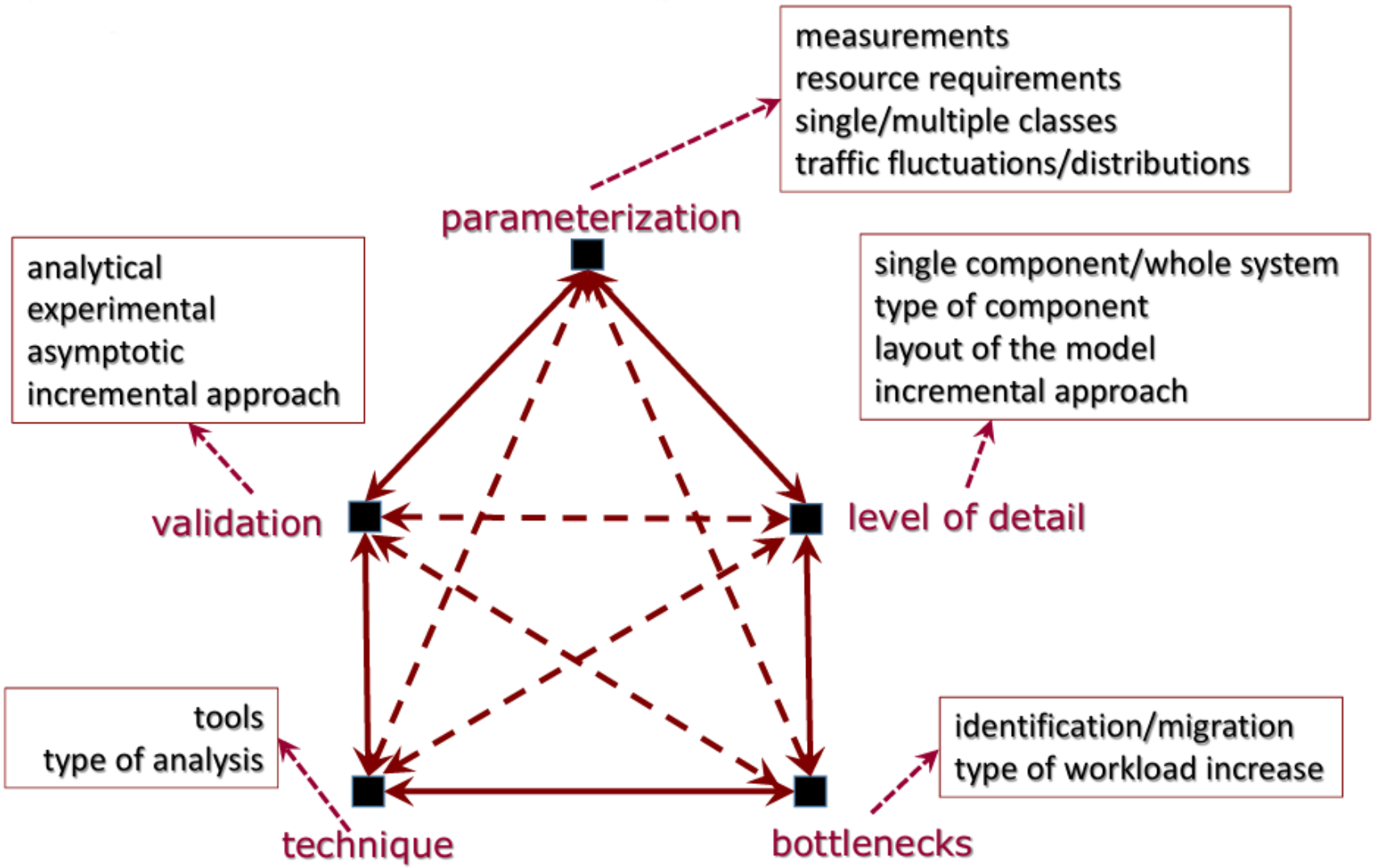
Stations executing *Tasks*

Routing of *Jobs*

Interconnections between Stations

Phases in model construction

The definition of a model follows several steps:

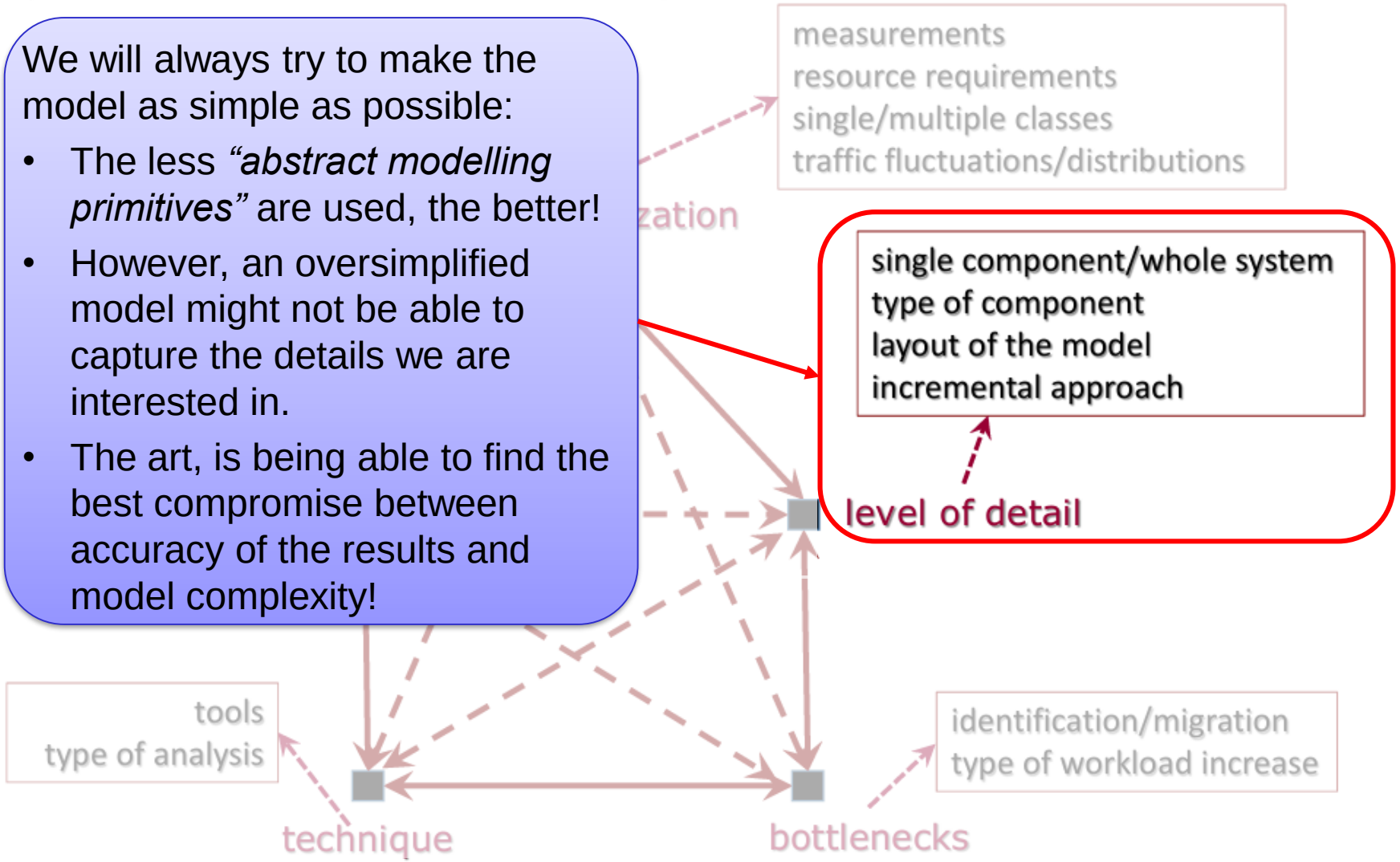


Phases in model construction

Level of detail:

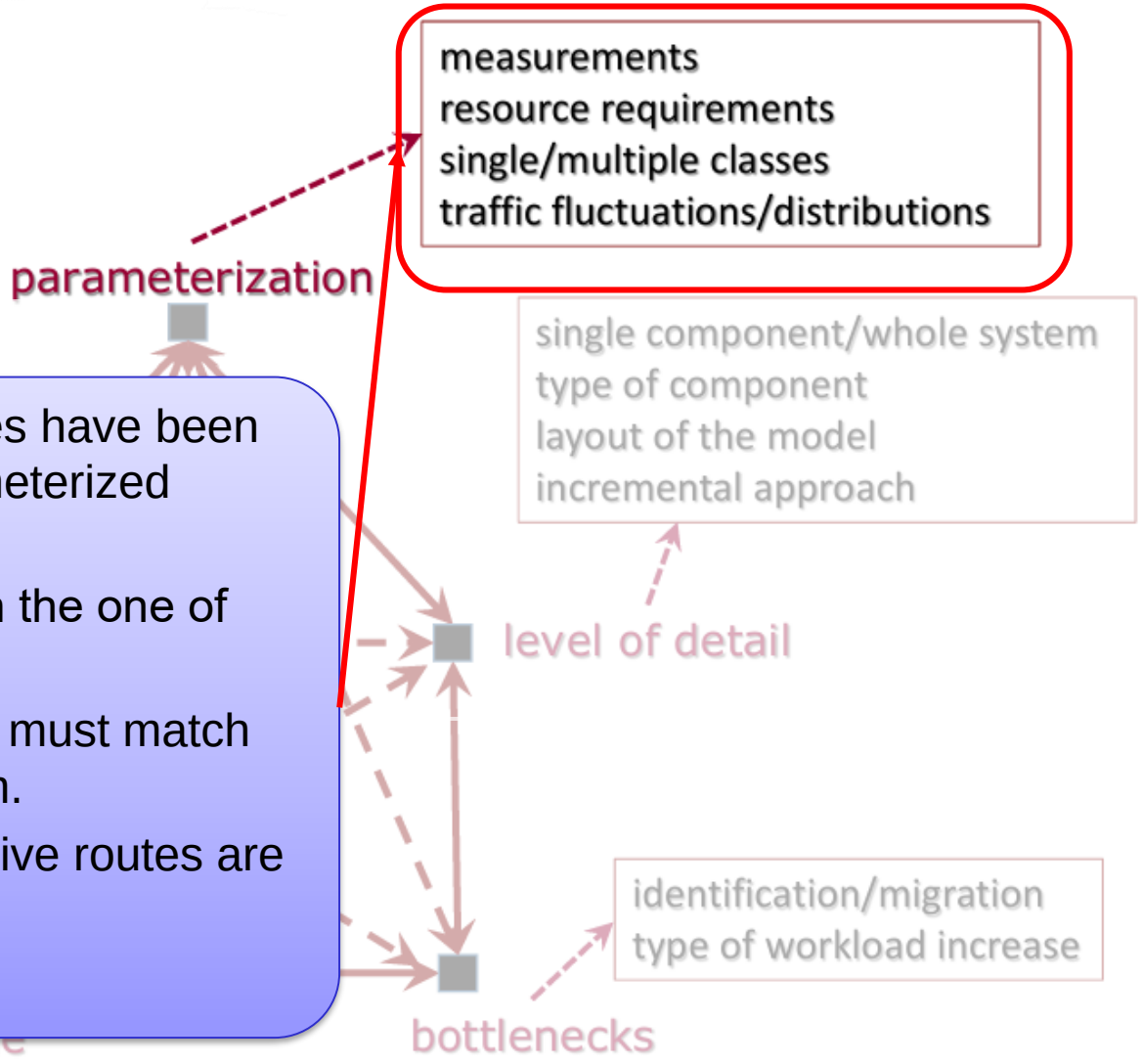
We will always try to make the model as simple as possible:

- The less *"abstract modelling primitives"* are used, the better!
- However, an oversimplified model might not be able to capture the details we are interested in.
- The art, is being able to find the best compromise between accuracy of the results and model complexity!



Phases in model construction

Parmeterization:

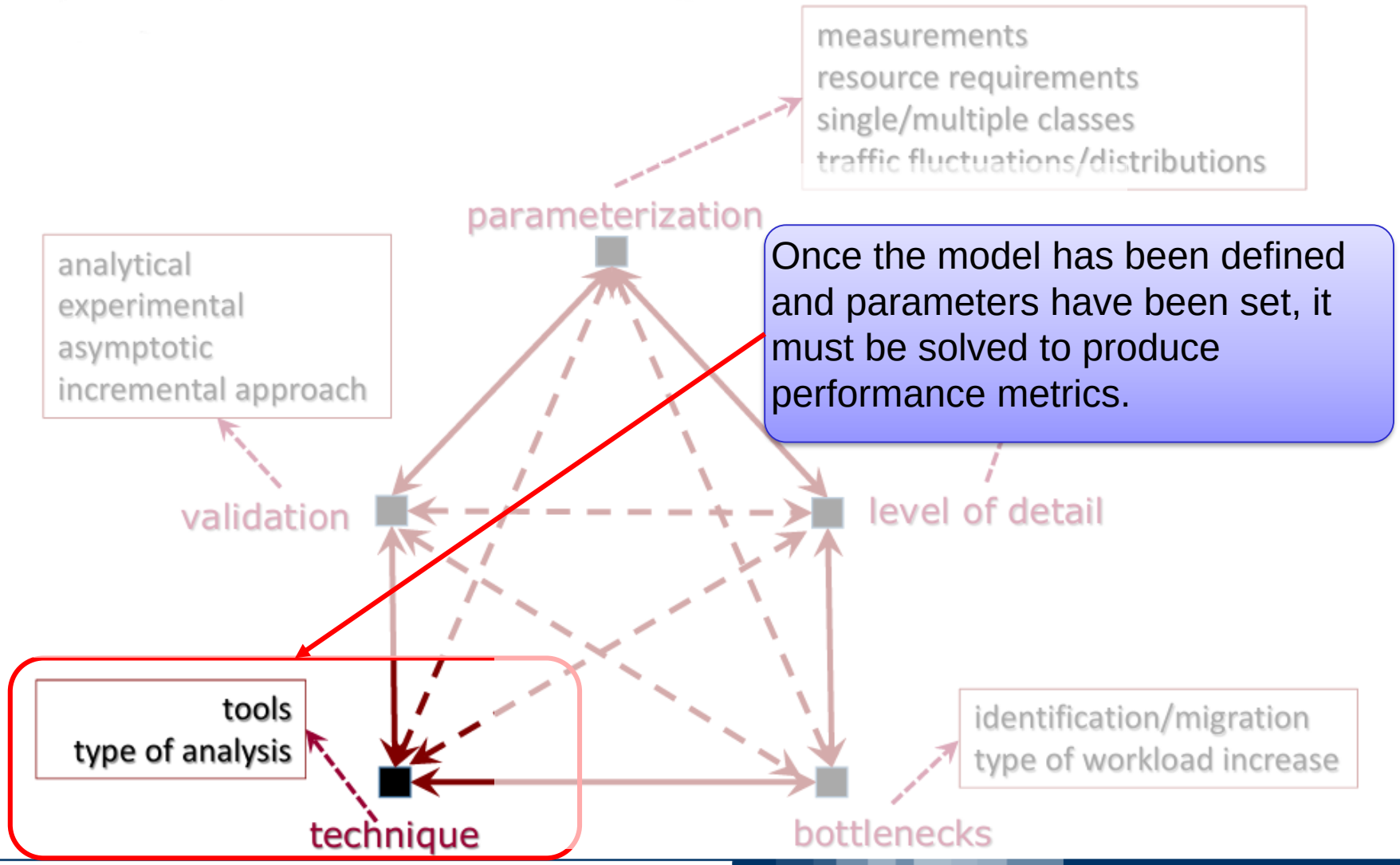


Once the modelling primitives have been chosen, they must be parameterized according to real system:

- Their “speed” must match the one of entities under study.
- The number of resources must match the one of the real system.
- The way in which alternative routes are chosen must match
- … and so on!

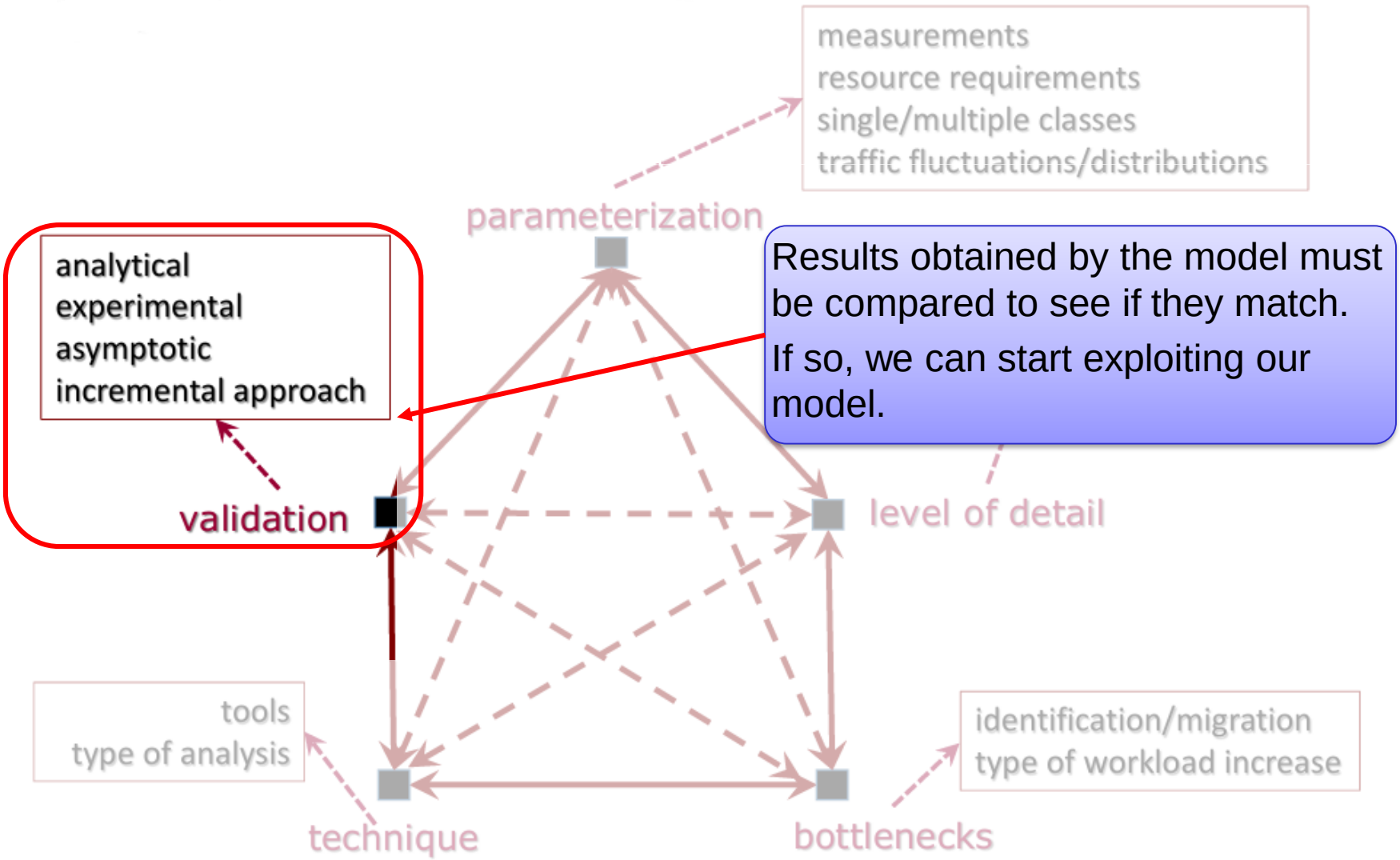
Phases in model construction

Technique:



Phases in model construction

Validation:

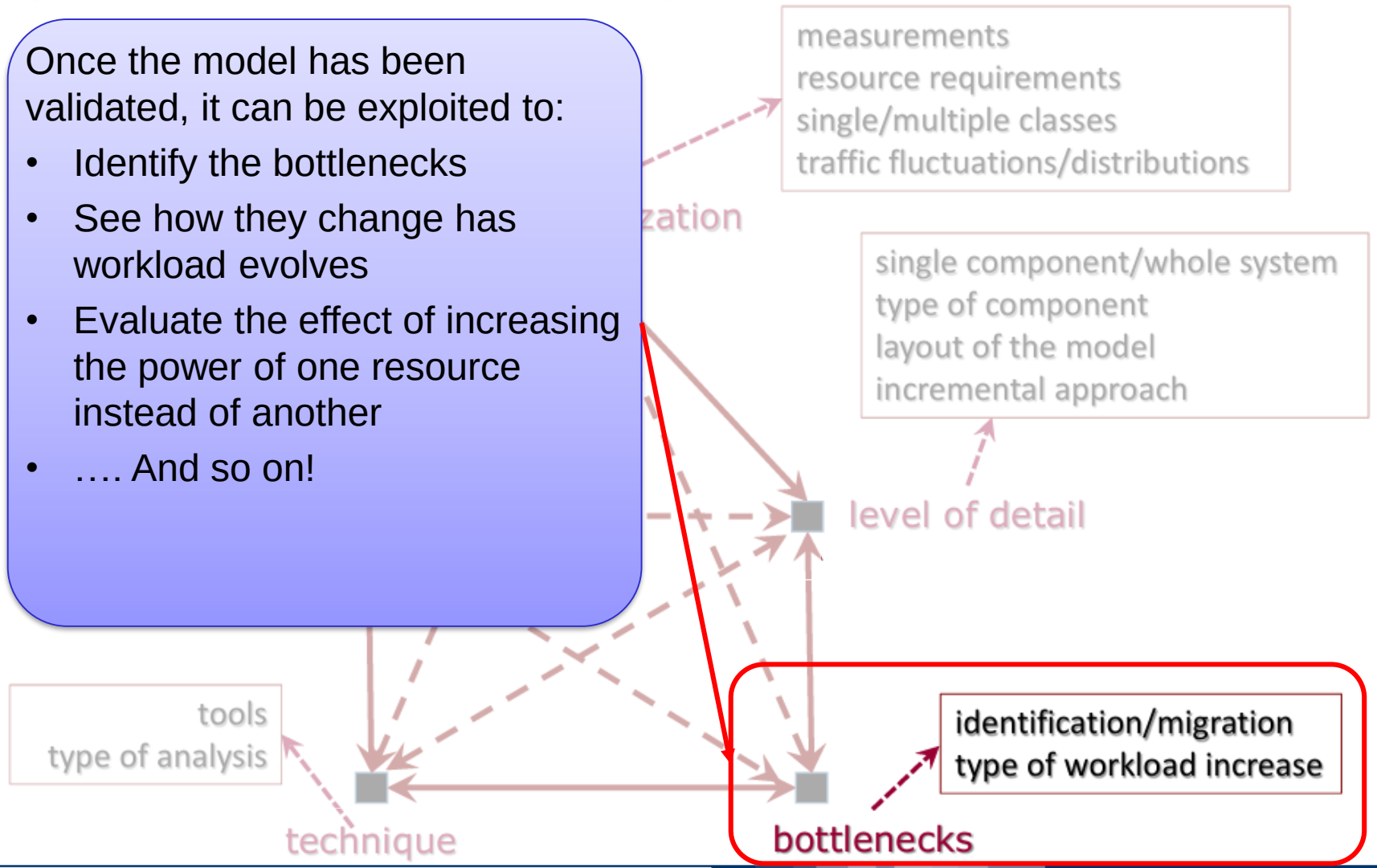


Phases in model construction

Bottlenecks:

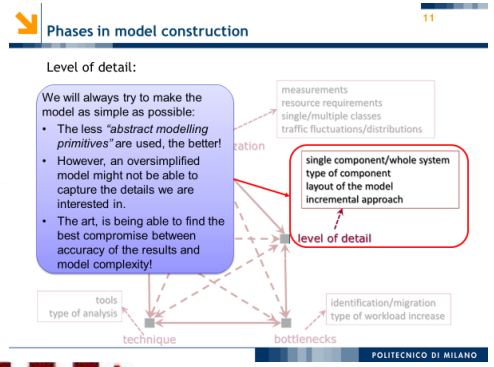
Once the model has been validated, it can be exploited to:

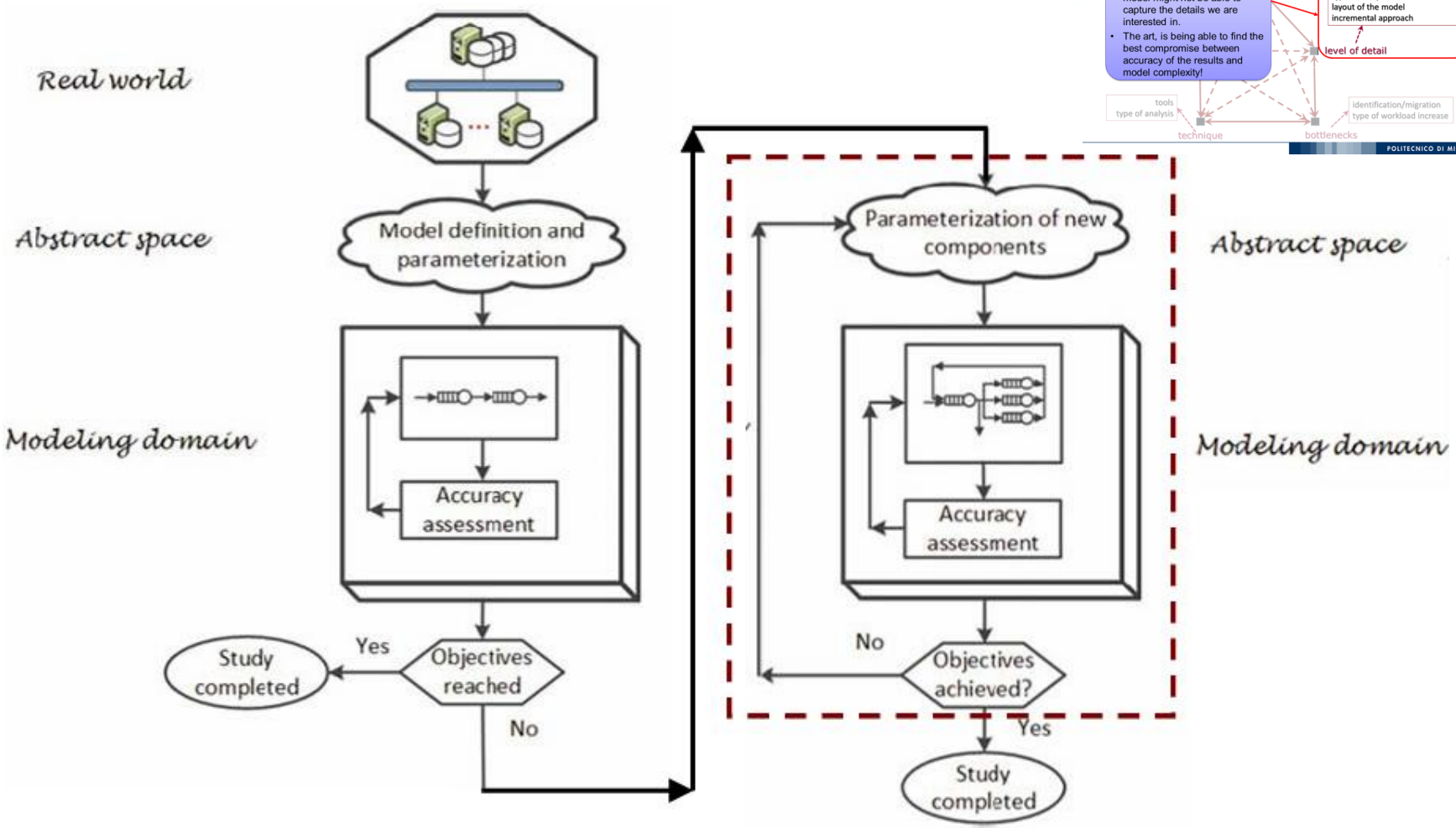
- Identify the bottlenecks
- See how they change has workload evolves
- Evaluate the effect of increasing the power of one resource instead of another
- …. And so on!



Incremental approach

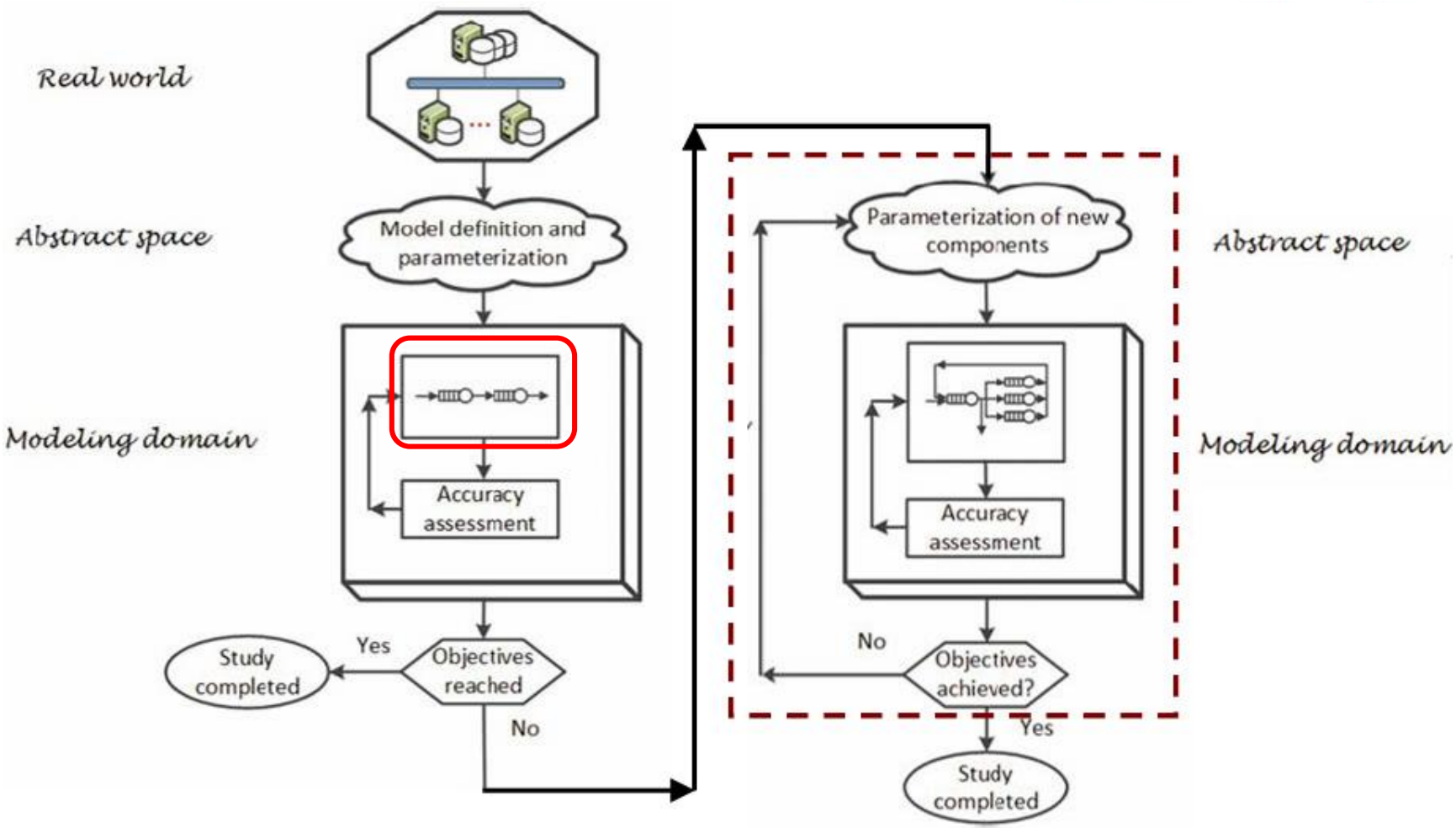
An incremental approach in cycles is followed in refining the model until the goals are reached.





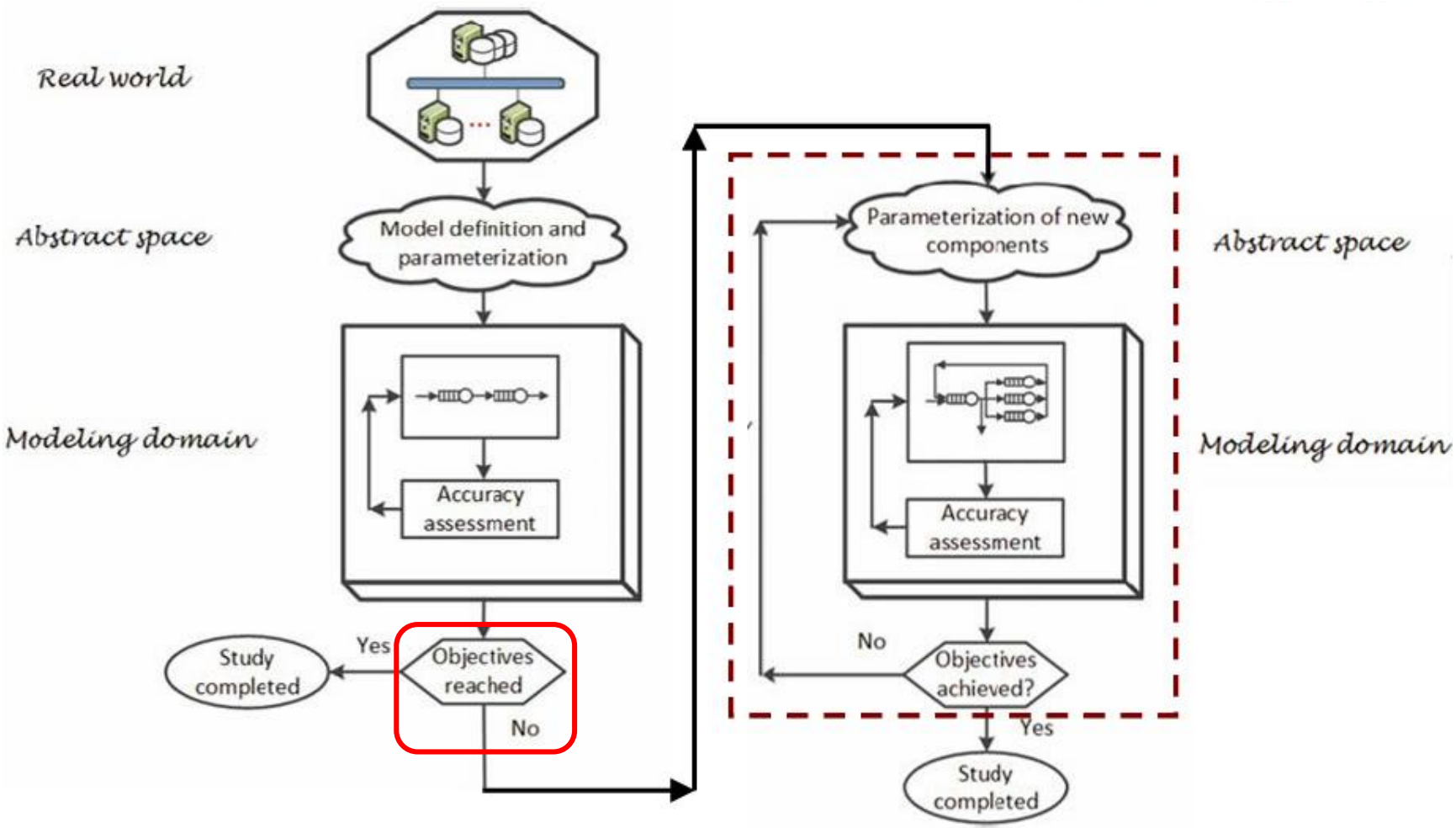
Incremental approach

We can, in a Web Server model for example, start considering the storage as a single component.



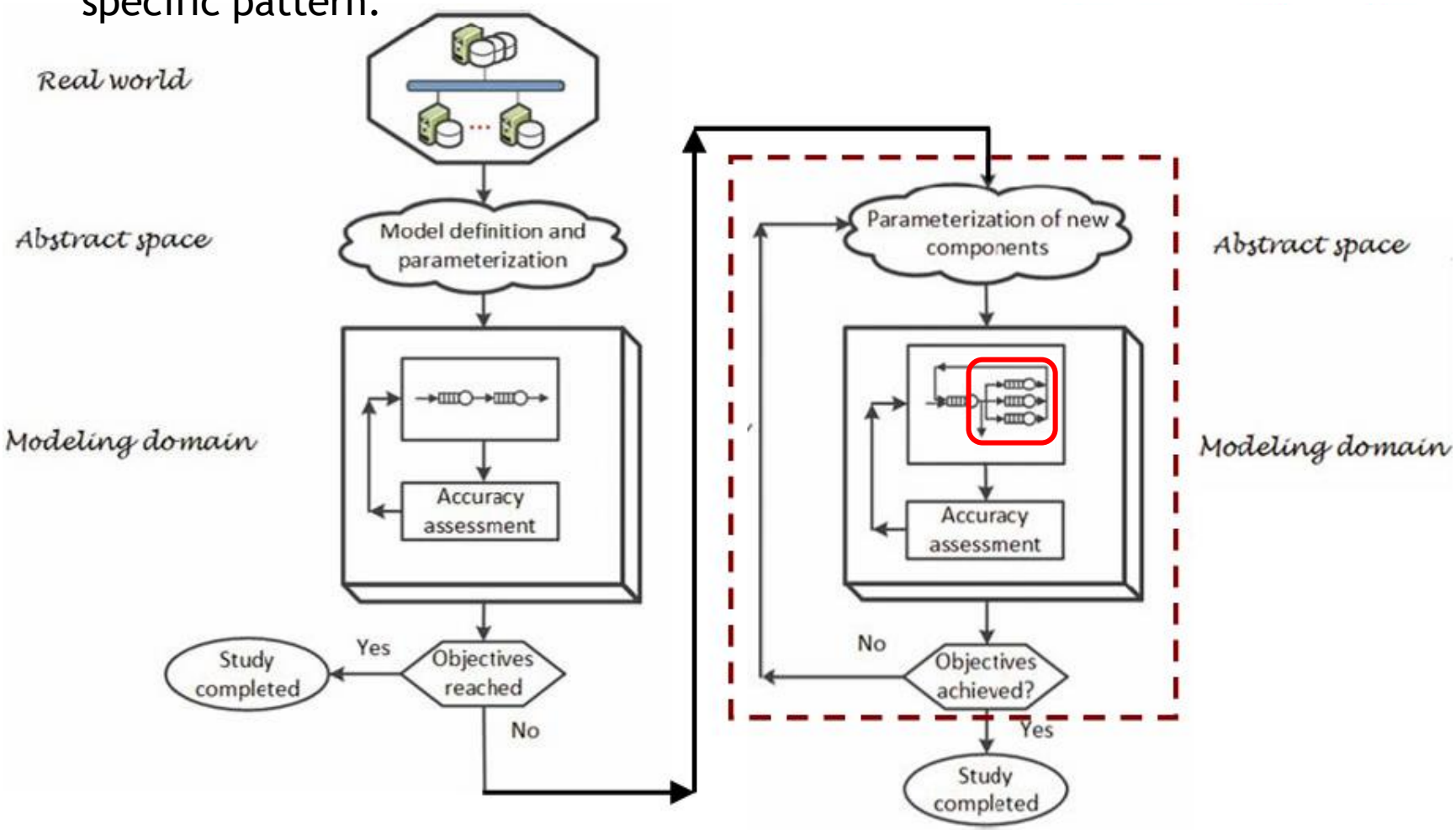
Incremental approach

If we fail the validation phase, by analyzing the differences we can understand that the problem is that the storage does not behave as modelled.



Incremental approach

We can enhance the model by better characterized the storage, considering that it is composed, for example, by different devices, used according to a specific pattern.



Performance indices

Performance indices measure the ability of the system to perform its task.

Workload accounts for the difficulty, length and number of tasks that have to be performed.

APPENDIX D — FINA TABLE OF DEGREES OF DIFFICULTY

This table became effective on September 15, 2009

New dives and dives which have been changed are shaded.

SPRINGBOARD		ONE METER				THREE METER			
		STR	PIKE	TUCK	FREE	STR	PIKE	TUCK	FREE
Forward Group		A	B	C	D	A	B	C	D
101	Forward Dive	1.4	1.3	1.2	-	1.6	1.5	1.4	-
102	Forward Somersault	1.6	1.5	1.4	-	1.7	1.6	1.5	-
103	Forward 1½ Somersaults	2.0	1.7	1.6	-	1.9	1.6	1.5	-
104	Forward 2 Somersaults	2.6	2.3	2.2	-	2.4	2.1	2.0	-
105	Forward 2½ Somersaults	-	2.6	2.4	-	2.8	2.4	2.2	-
106	Forward 3 Somersaults	-	3.2	2.9	-	-	2.8	2.5	-

Performance indices

The description of a system component includes parameters characterizing its workload, and performance indices that can be estimated. The most important are:

Workload characterization:

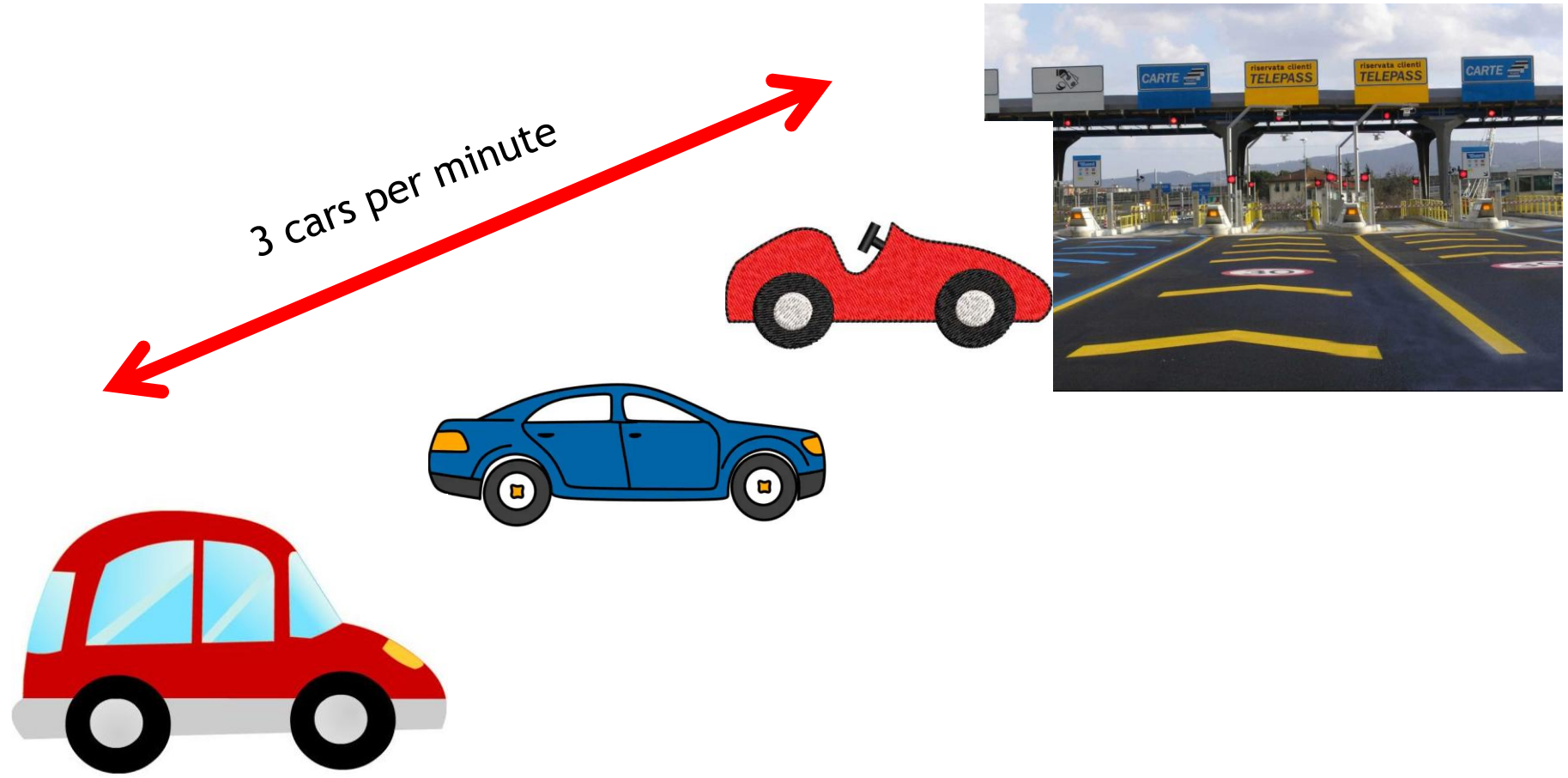
- *Arrival rate*
  - *(Average) Inter-arrival time*
- *(Average) Service time*

Performance indices:

- *Utilization*
- *(Average) Response time*
- *(Average) Queue length*
- *Throughput*

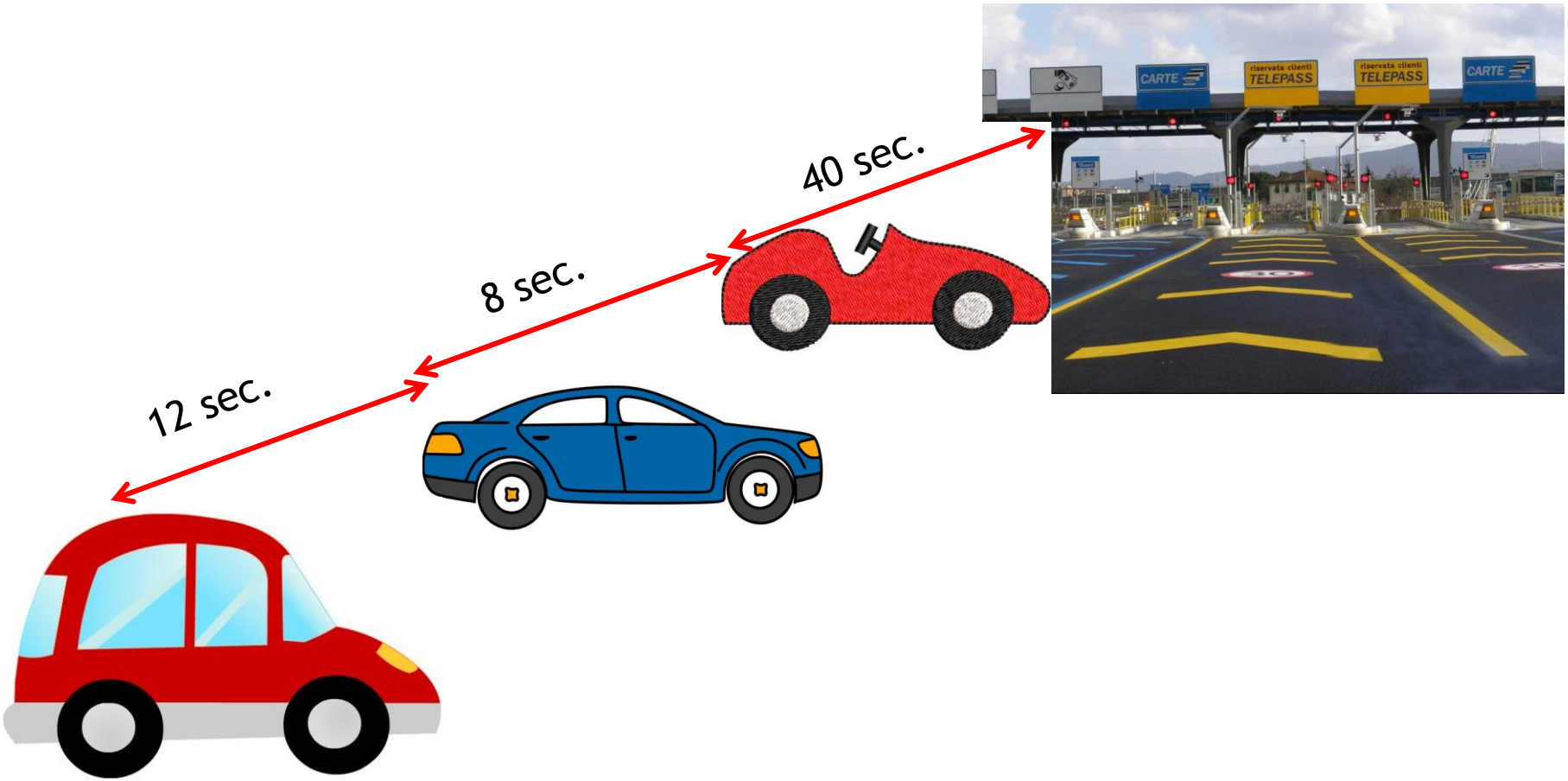
Workload characterization

The *arrival rate* λ is the frequency at which jobs arrives at a given station.



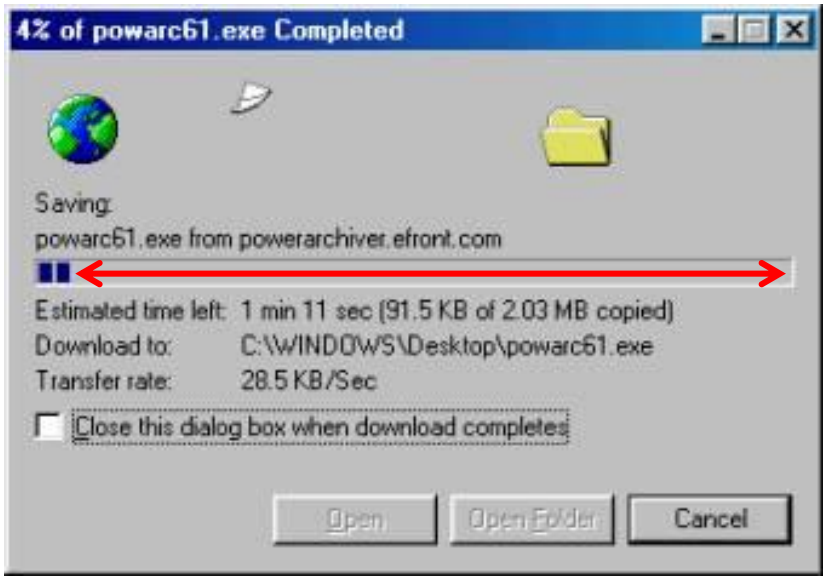
Workload characterization

The *inter-arrival time* a_i, measures the time between two consecutive arrivals (the *i*-th and *i*-th+*1*) to the system: as we will see, it is closely related to the *arrival rate* just introduced.



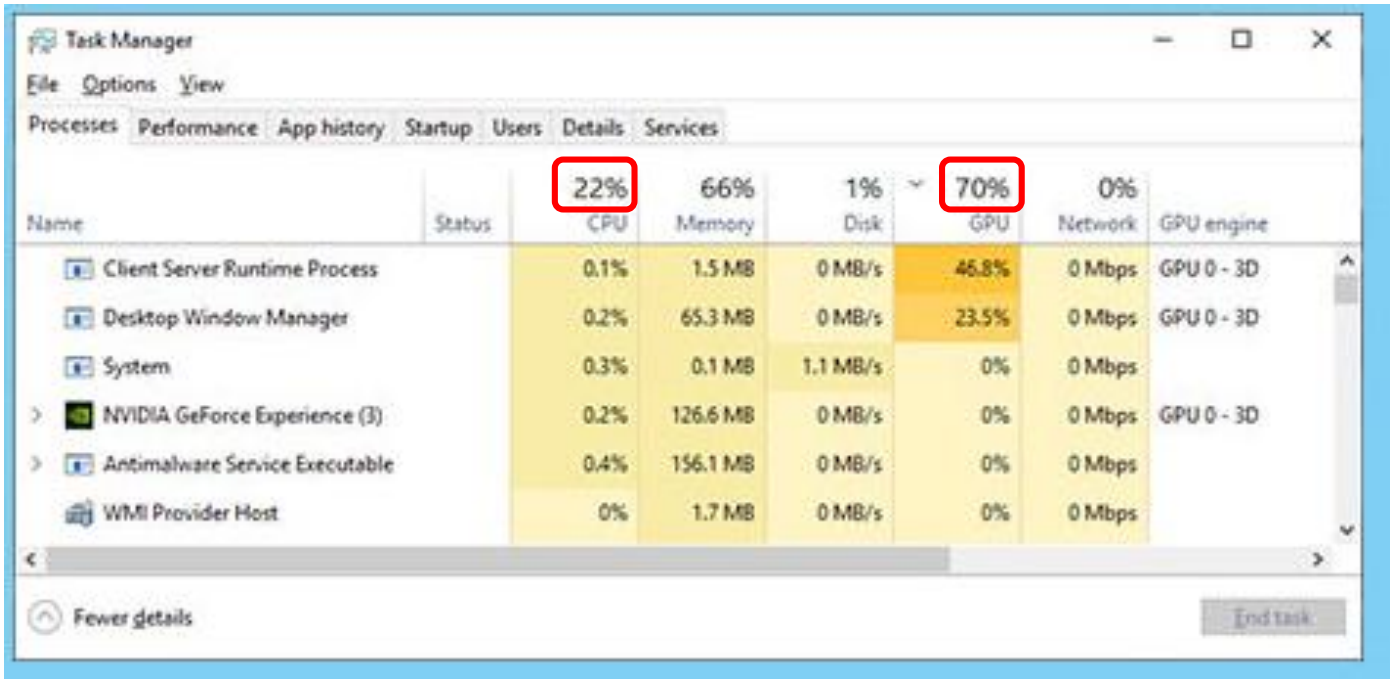
Workload characterization

The *service time* s_i is the time required by the *i*-th job to complete its service.



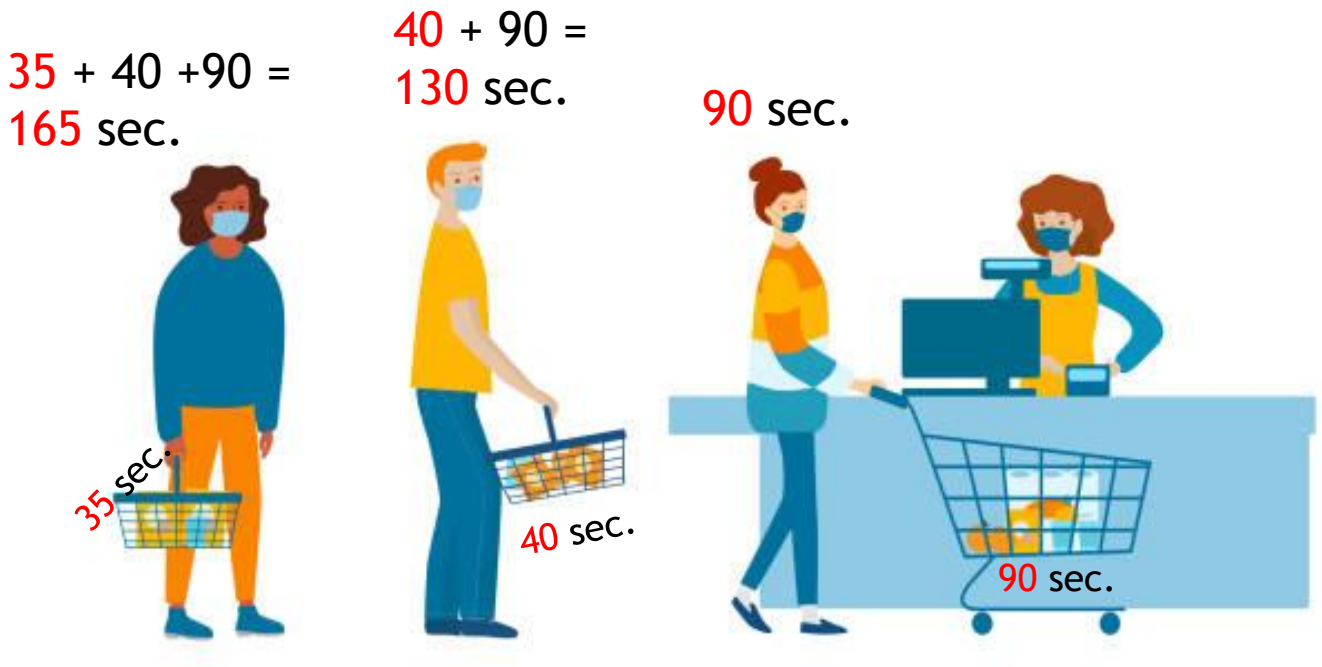
Performance indices

The *utilization U* is the fraction of time a server is busy (not idle while waiting for a new job to arrive).



Performance indices

The *response time* r_i is the time spent by the *i*-th job at a service center, including service and queuing time.



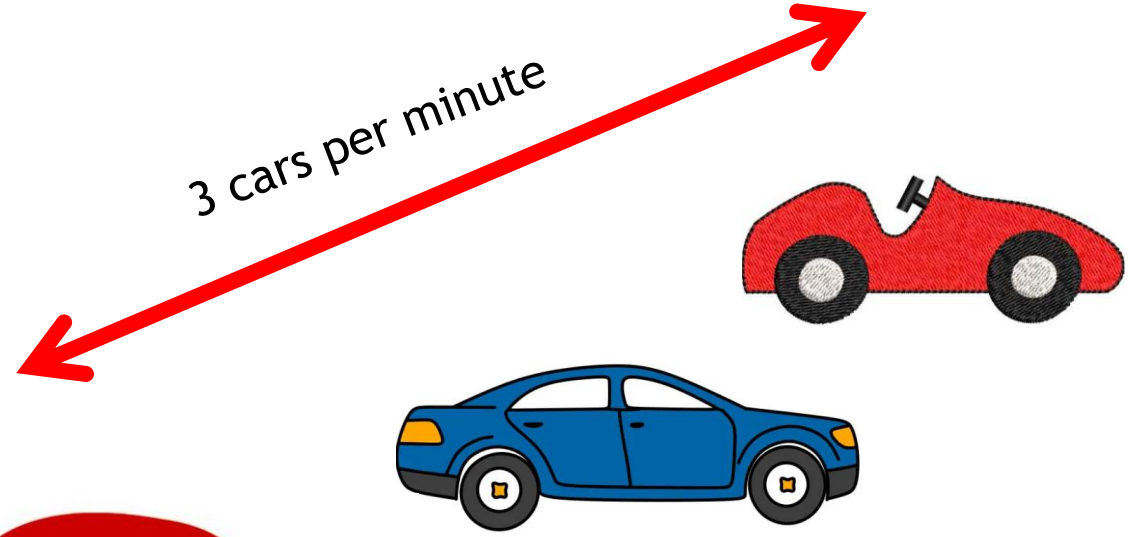
(supposing all costumers arrive at the same time at an empty counter)

Performance indices

The *queue length N(t)* accounts for the number of jobs in a service station (both the ones being served and the ones in the queue), at a given point in time *t*.

Performance indices

The *throughput X* describes the rate at which jobs are served and depart from the station.



Average values

Utilization *U*, Arrival rate λ, and Throughput *X* are *long run measures*: they are meaningful only when considering a *sufficiently* long amount of time where the system exhibits a *similar behavior*.

Sufficiently long is relative to the application: for the utilization, it could be even as short as one second, and for the throughput of a production line as long as one year.

Similar behavior is more difficult to define, and can include different time scales and oscillations. In most of the cases (but not limited to this), it means that workload is *constant*, or it follows a *specific statistical pattern* (but then the difficulty is defining what a "*specific statistical pattern*" means).

Average values

Number of jobs $N(t)$, inter-arrival times a_i, service times s_i, and response times r_i, are instead time or job dependent measures.

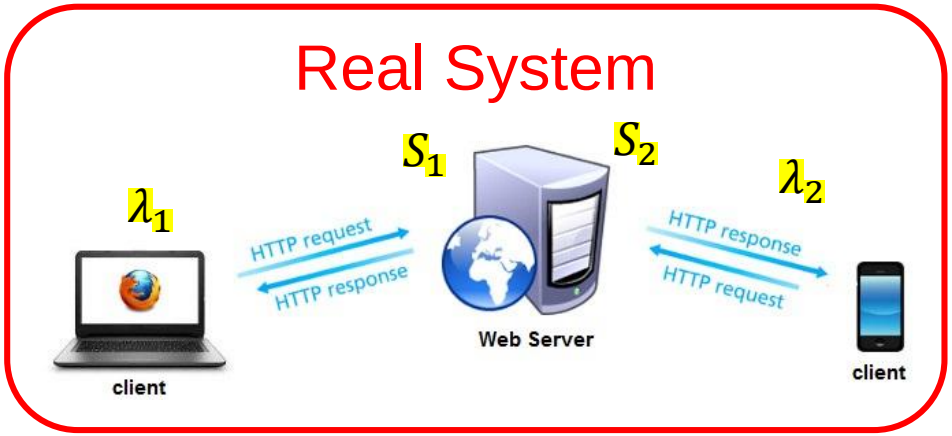
In most of the cases we are interested in the average of such quantities, with the average computed in the same time interval discussed for U, λ, X. These measures are:

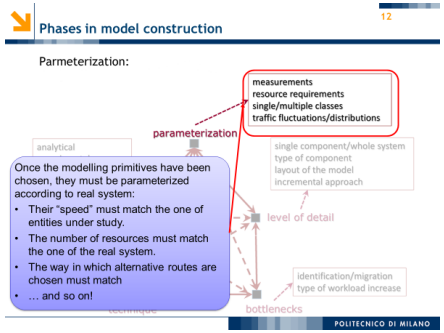
- Average number of jobs: N
- Average inter-arrival time: $\bar{A}$
- Average service time: S
- Average response time: R

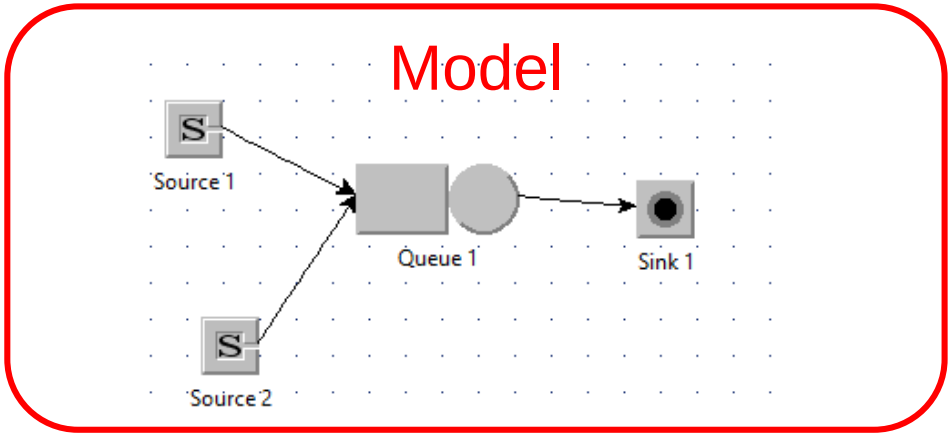
To simplify the discussion, in the following we will only focus on a given interval T, when this interval T tends to the infinity.

Performance indices and workloads: model and reality

The workload, such as arrival rate and average service time, are measured on the real system.

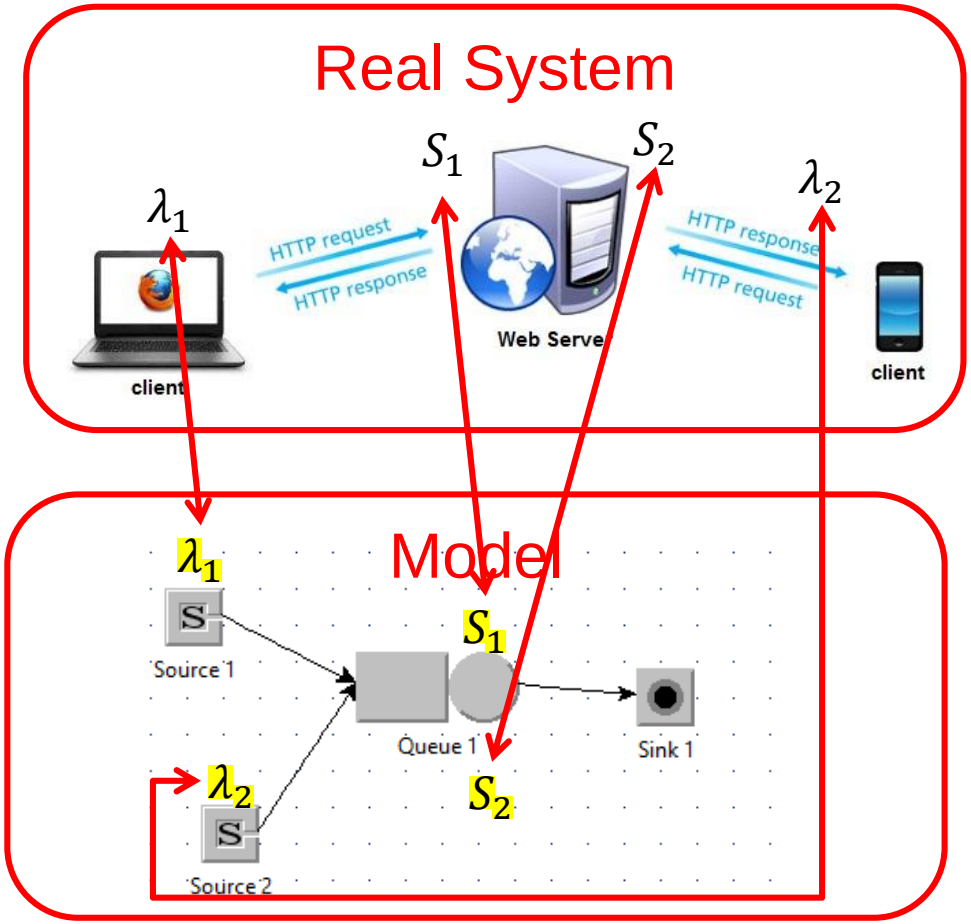






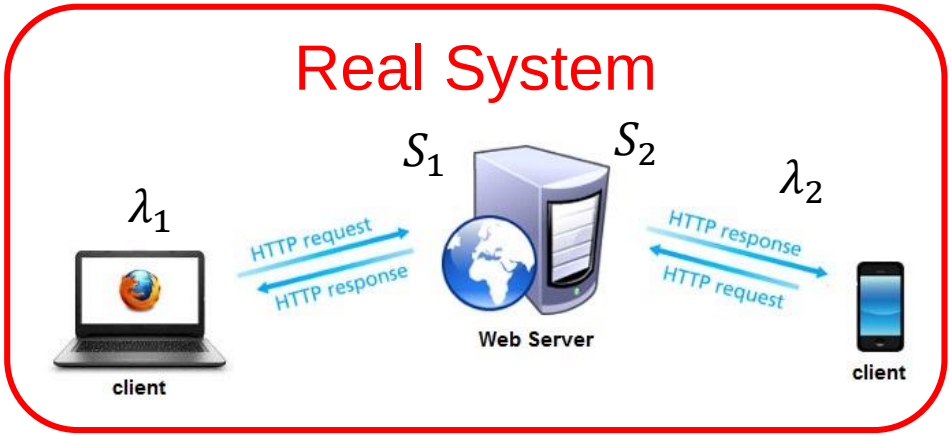
Performance indices and workloads: model and reality

They are then used as the input of a model.

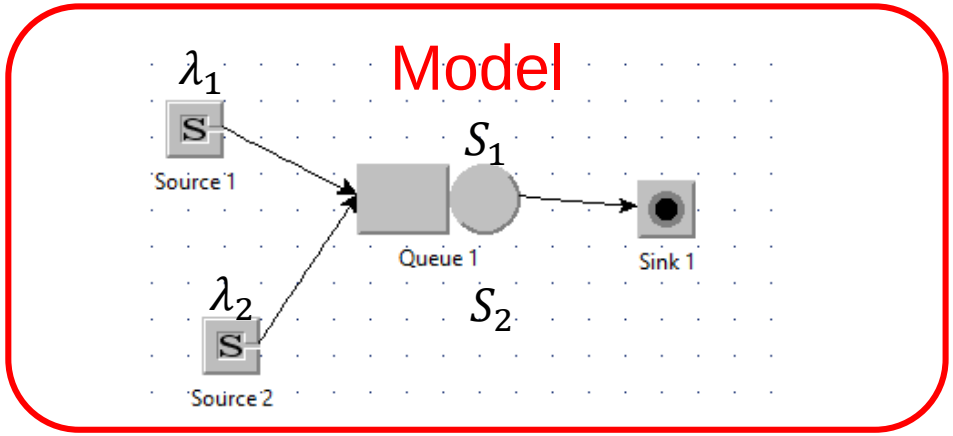


Performance indices and workloads: model and reality

Performance indices are measured both on the real system being considered and its model.



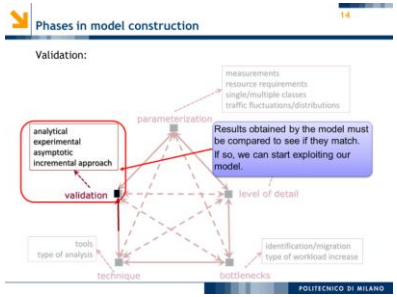
R_{Sys} U_{Sys} X_{Sys}

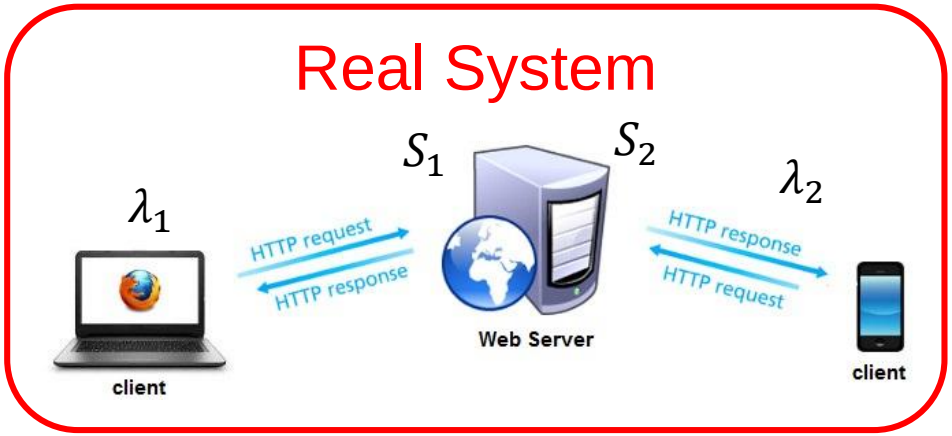


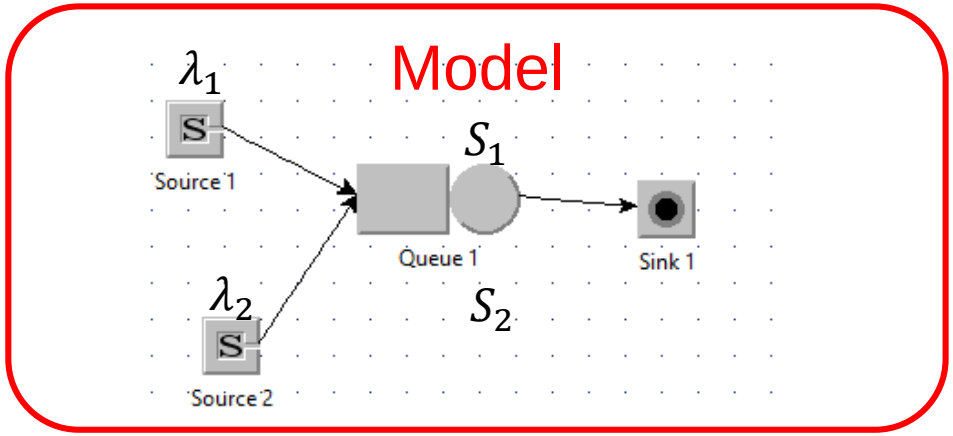
R_{Model} U_{Model} X_{Model}

Performance indices and workloads: model and reality

Indices derived from a model should match closely the ones measured on the corresponding real system: this check is what we introduced as *Validation*.







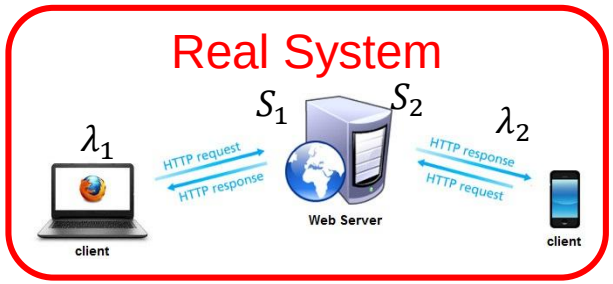
R_{Sys} U_{Sys} X_{Sys}

R_{Model} U_{Model} X_{Model}

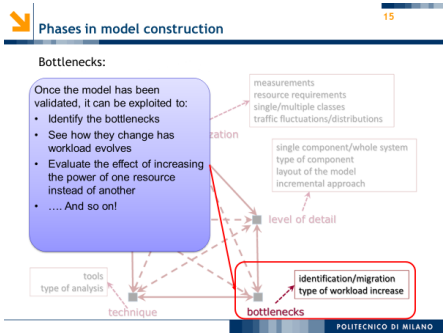
In most cases, average value will be enough to provide a good system description. In other situations, we will need a more detailed description of both performance indices and workloads

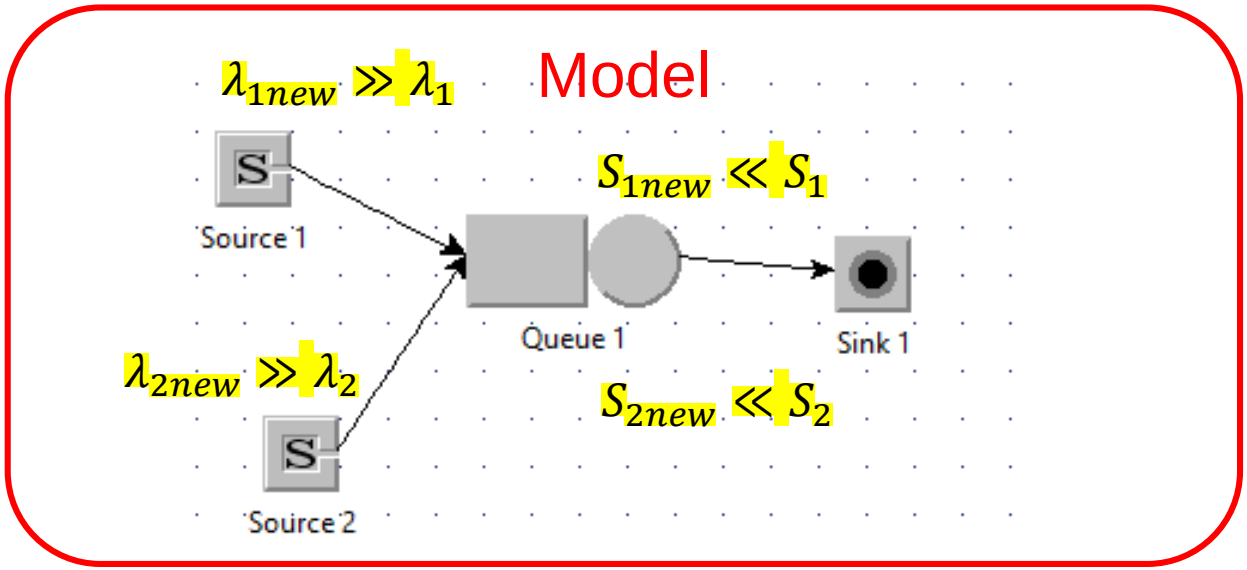
Model exploitation

Once the model has been validated with the considered workload, it is studied varying arrival rates, service times, and other configuration parameters to see their effects on the performance indices.



This is the interesting part of the job: using models to address the best improvements to be performed, planning them, and implement them to achieve specific goals.





R_{New}

U_{New}

X_{New}