A Simulation Model for Studying Cloud Applications through Software Architecture Evaluation

- Ph.D. Thesis -

María Julia Blas

- Advisors -

Dr. Horacio Leone - Dr. Silvio Gonnet

Software Product Quality

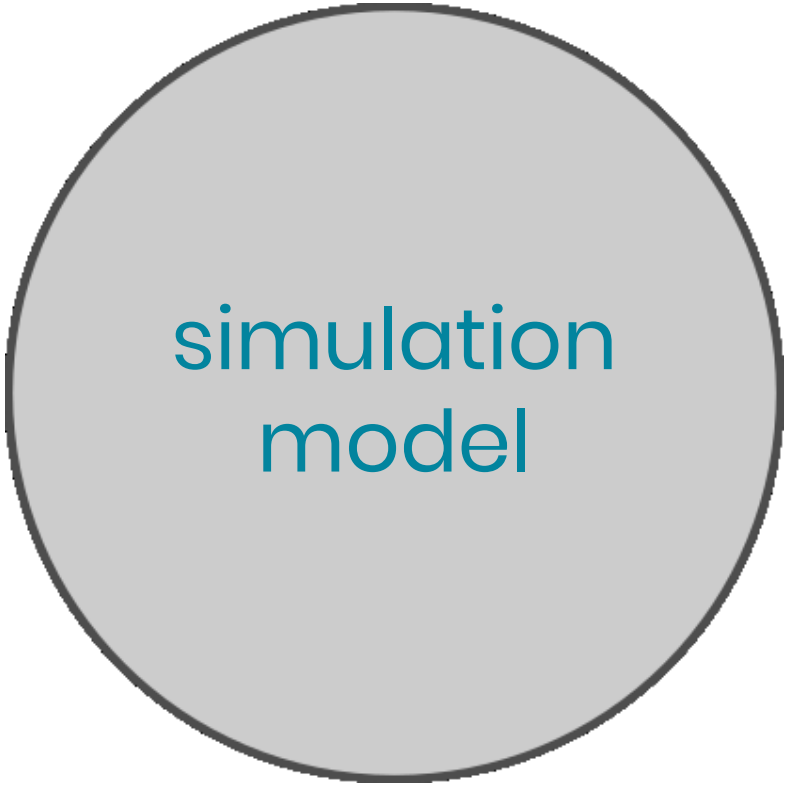
... refers to the degree to which software products meet their stated requirements.

Quality attributes should be achieved using well-known **architectural patterns**.

The **objective** of the thesis was to formulate a discrete event simulation model that allows the estimation of predefined quality attributes over the architecture design (i.e. the architecture evaluation) of web-based applications.

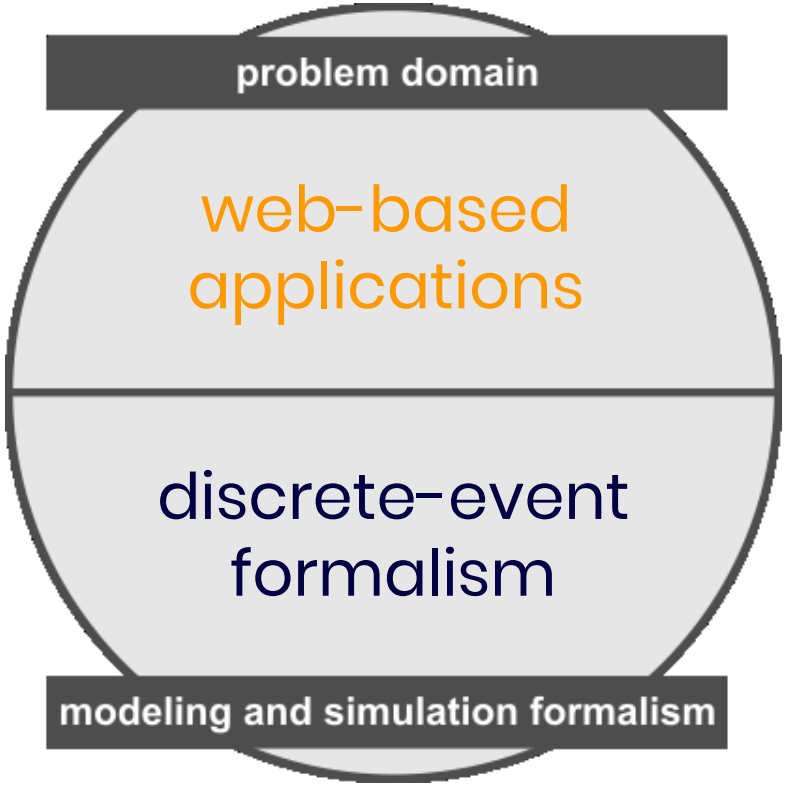
Thesis Contributions

Abstraction Level



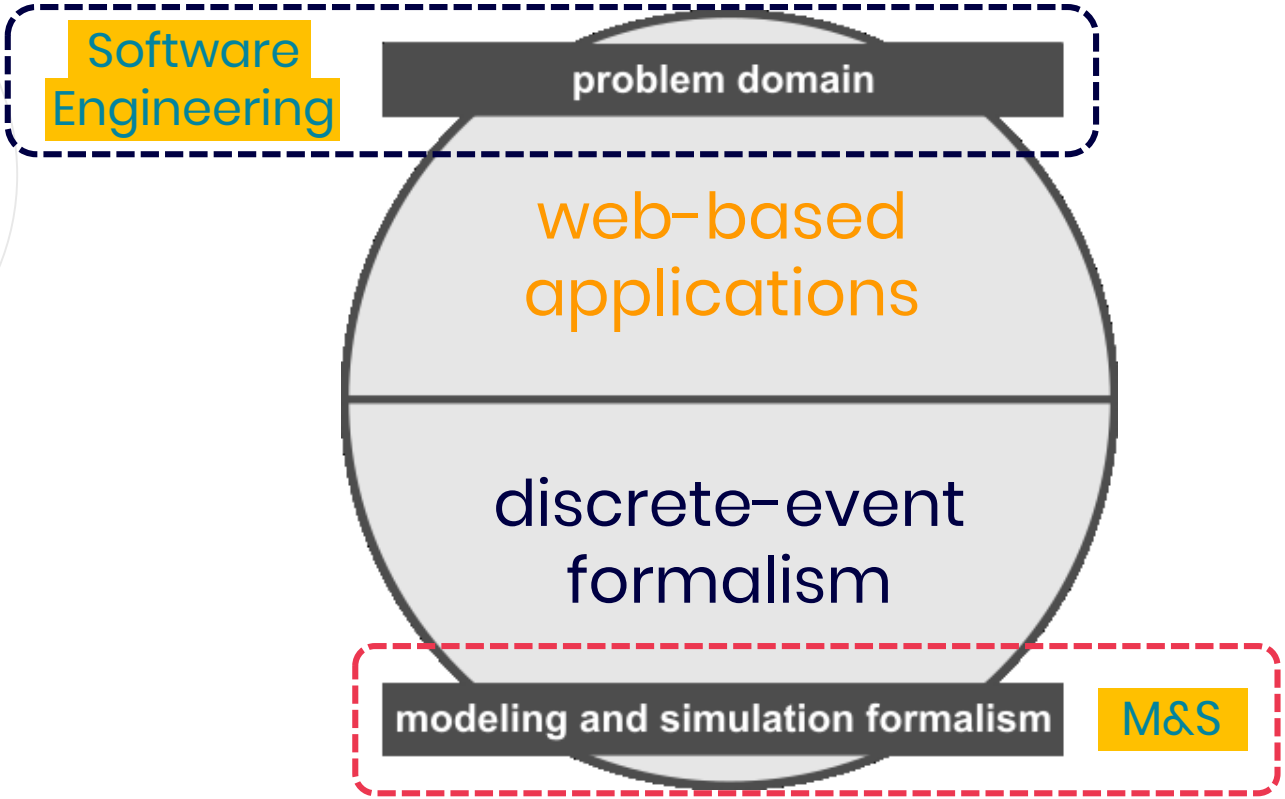
Thesis Contributions

Conceptualization Level



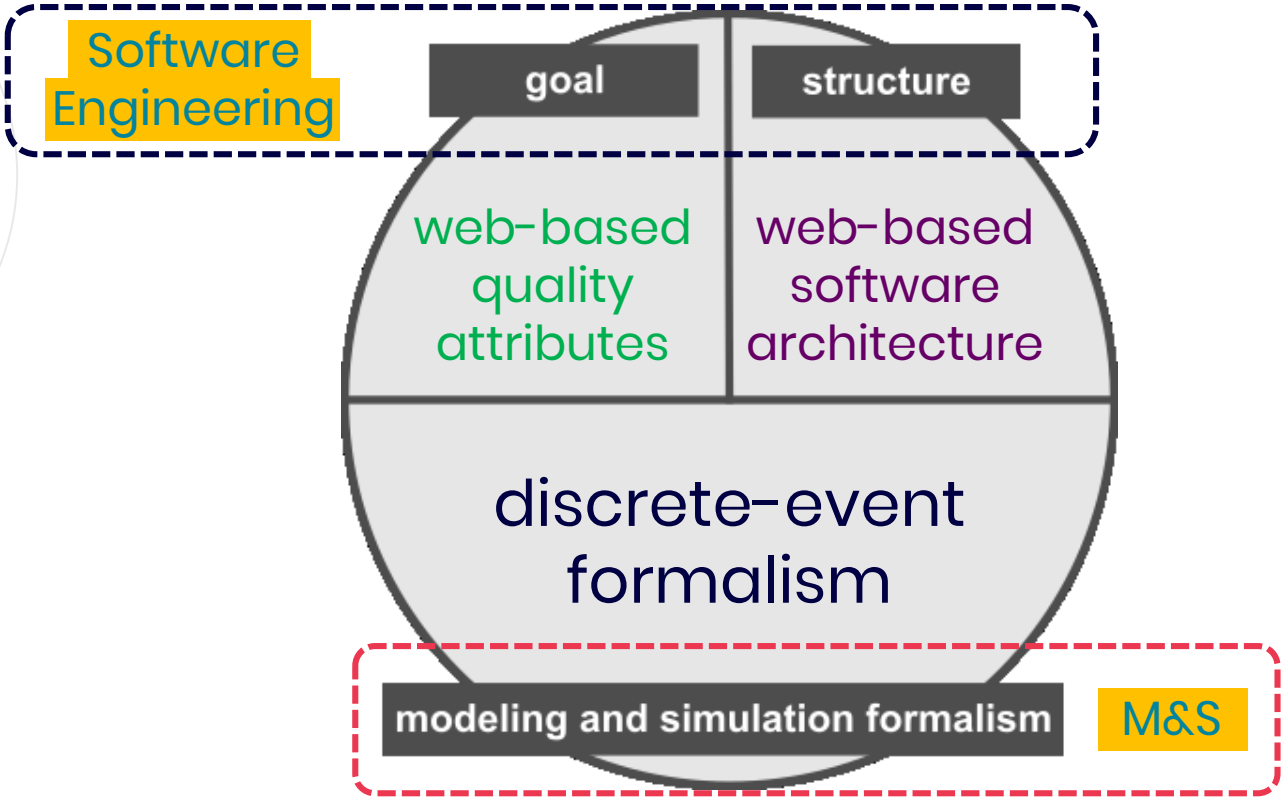
Thesis Contributions

Conceptualization Level



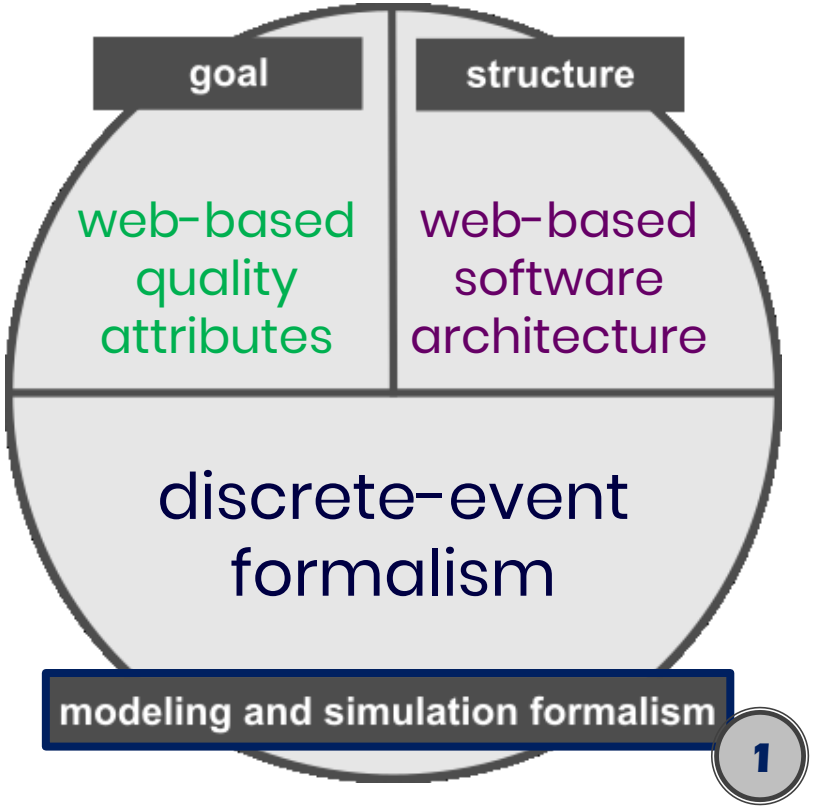
Thesis Contributions

Composition Level



Thesis Contributions

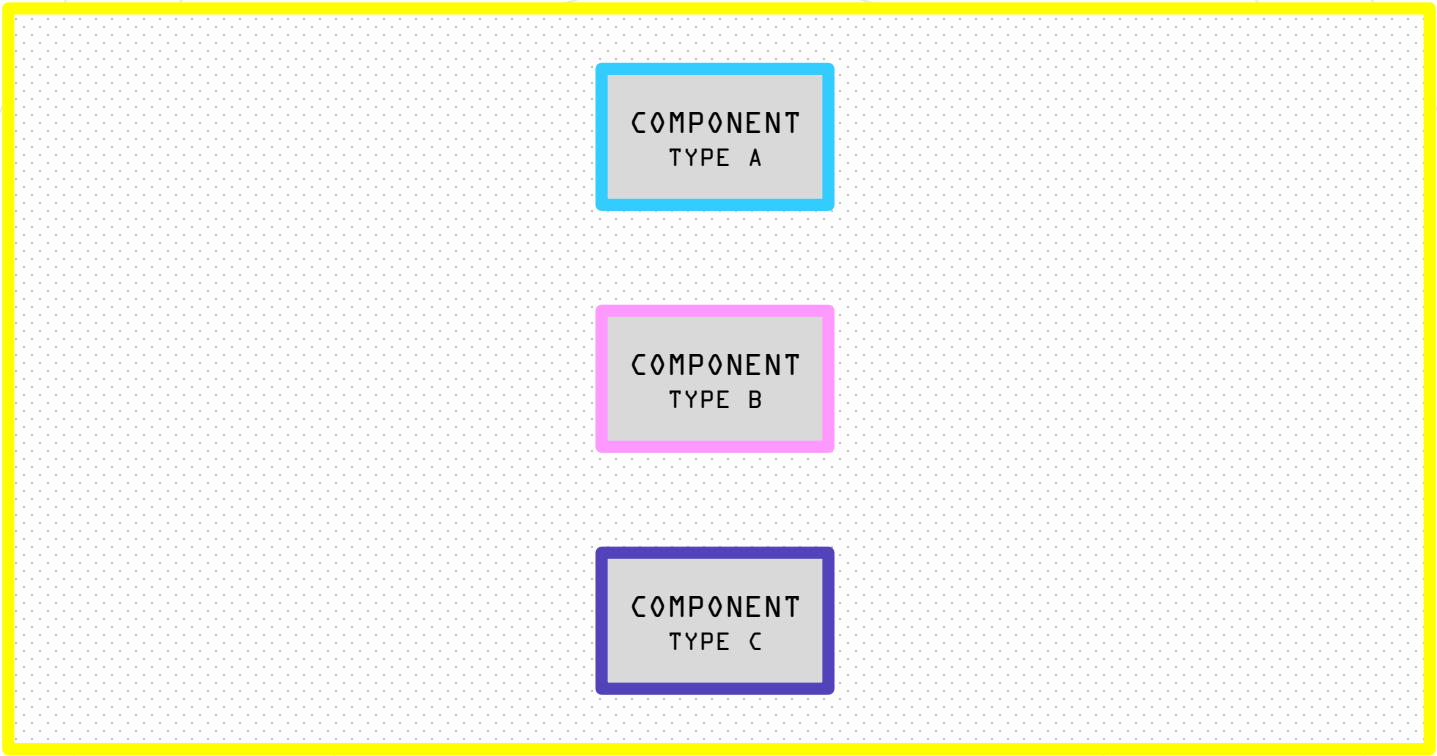
Composition Level



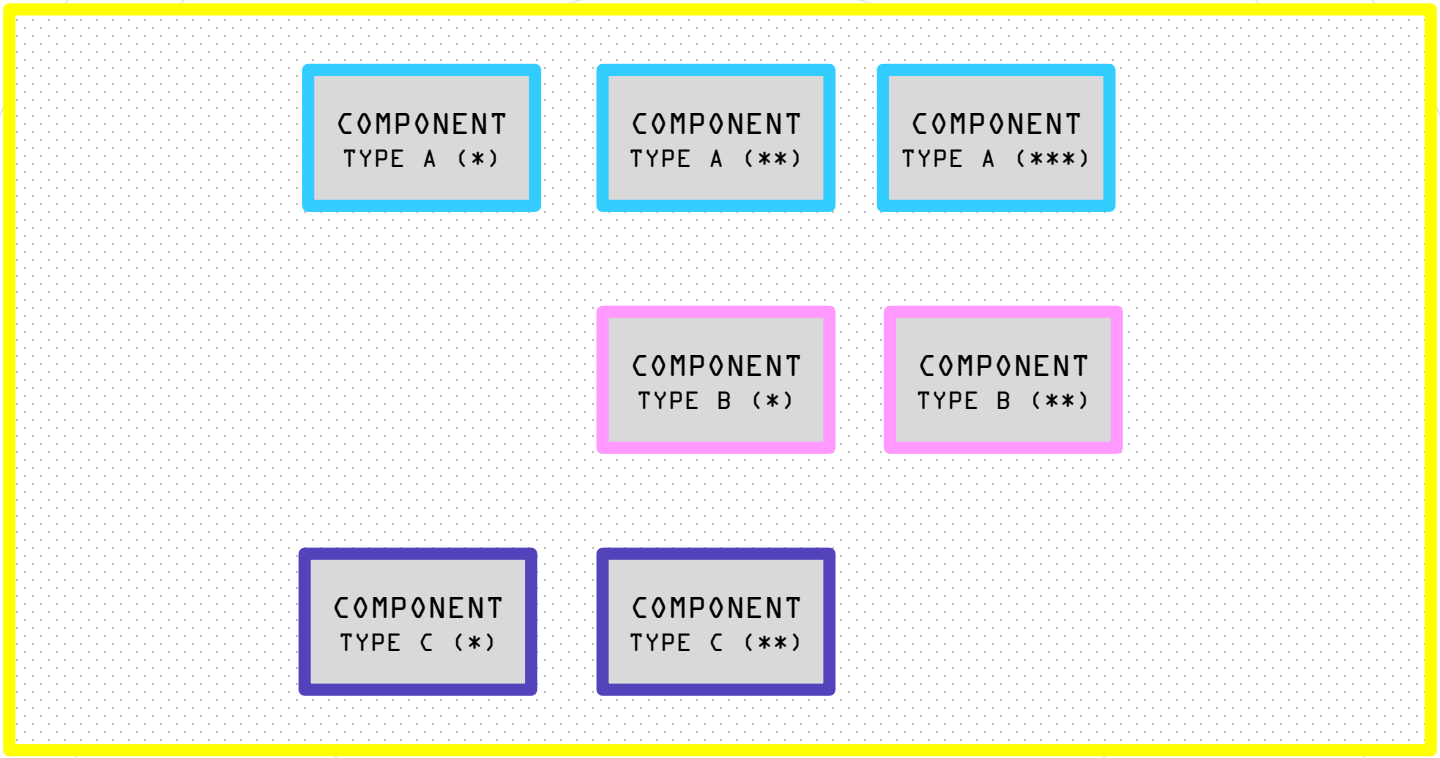
1

RDEVS

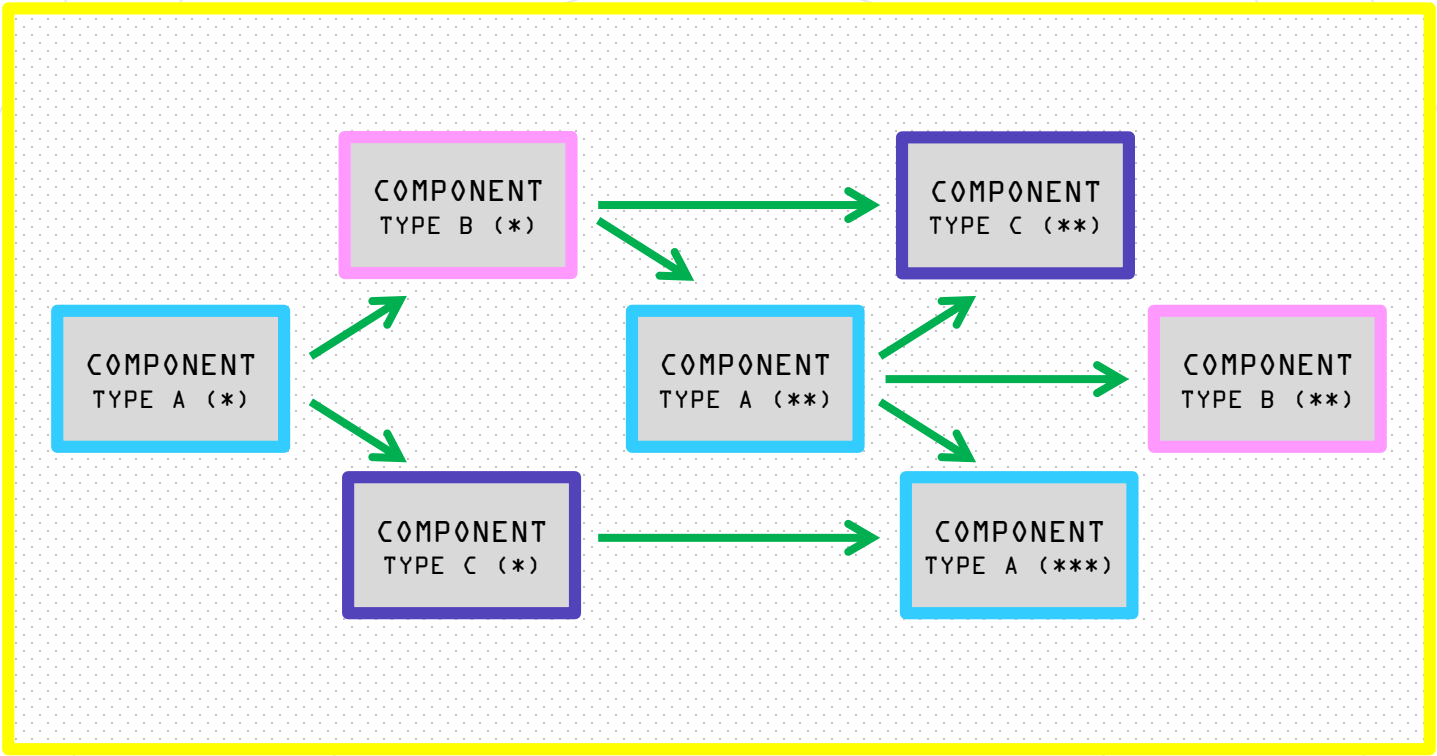
Routed DEVS



Routing Problem Definition



Routing Problem Definition



A **routing process** is defined as "*the part of a modeling scenario where the components need to interact among them by distinguishing the event sources and destinations in order to ensure their transference into the right model*".

Discrete Event System Specification (DEVS)

Atomic Model

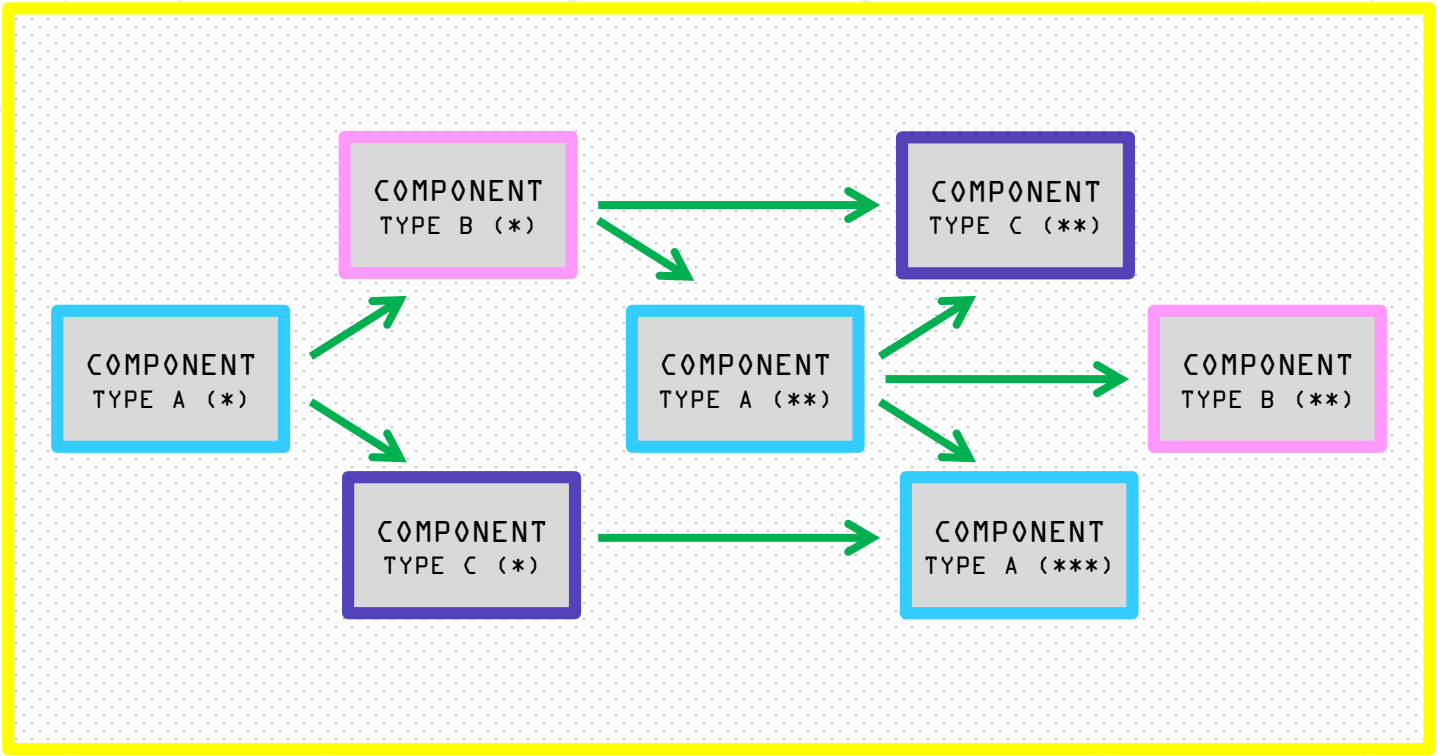
Describes a system as a sequence of deterministic transitions among states as well as how it reacts to external input events and how it generates output events.

(System) BEHAVIOR

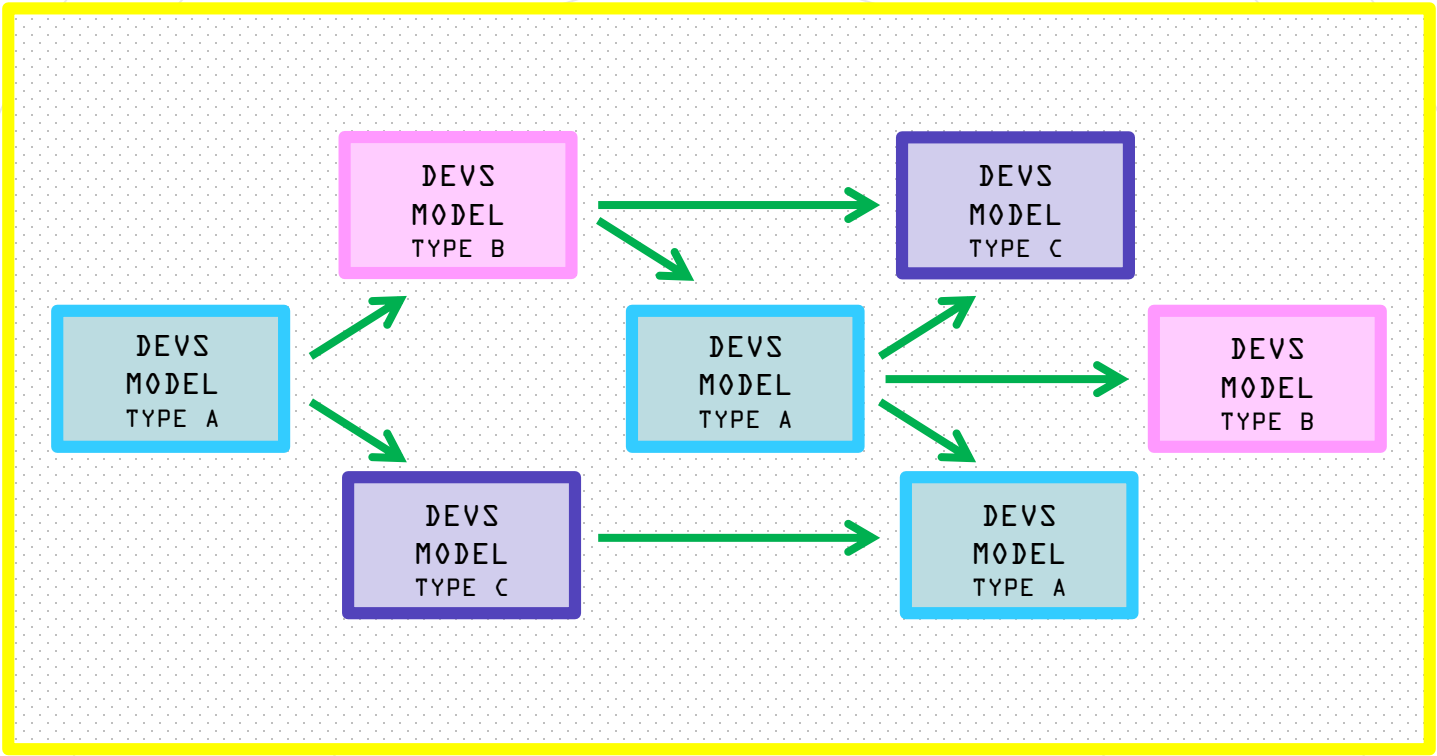
Coupled Model

Describes a system as a network of DEVS components (that can be either atomic DEVS or coupled DEVS).

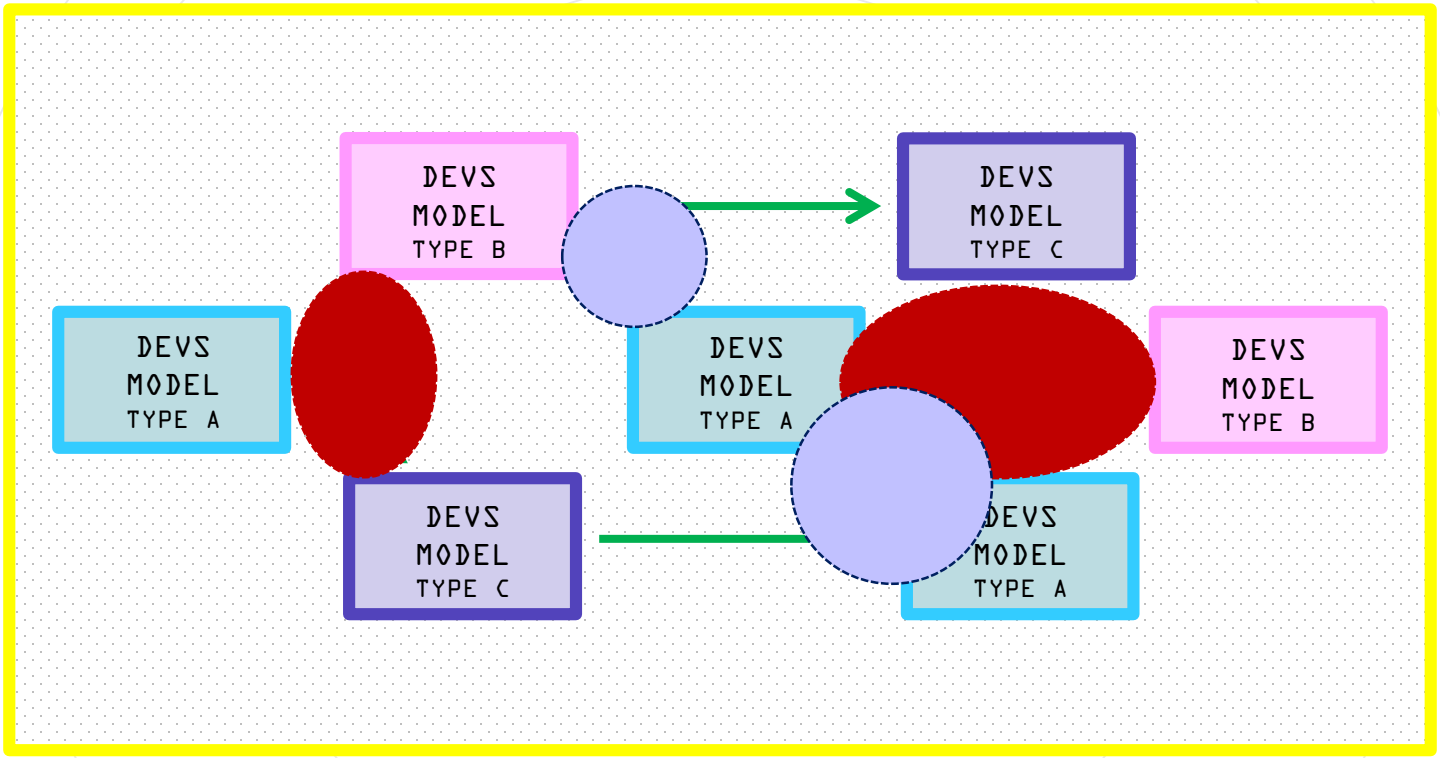
(System) STRUCTURE



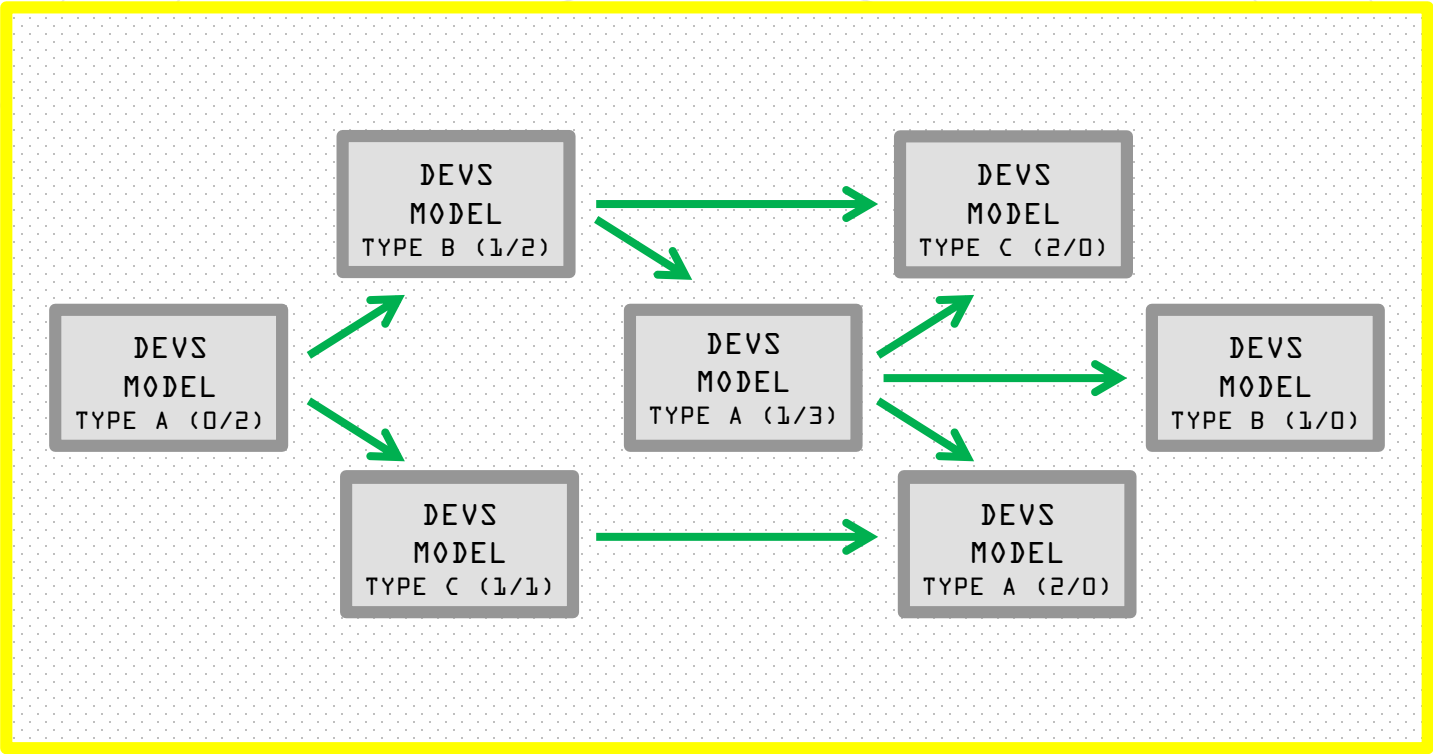
Routing Problem (DEVS Solution)



Routing Problem (DEVS Solution)



Routing Problem (DEVS Solution)



Routing Problem (DEVS Solution)

Routed DEVS

Routed Discrete Event System Specification *(RDEVS)*

The models use routing policies to authenticate senders/receivers prior executing their behavior with aim to provide an appropriate separation of concerns between the routing process description (i.e. structure and routing paths) and the behavior of components.

Routed DEVS

Routed Discrete Event System Specification *(RDEVS)*

The models use routing policies to authenticate senders/receivers prior executing their behavior with aim to provide an appropriate separation of concerns between the routing process description (i.e. structure and routing paths) and the behavior of components.

This allows using RDEVS formalism as a 'layer' above the DEVS formalism that provides routing functionality without requiring the user to 'dip down' to DEVS itself for any functions.

RDEVS Models

Essential Model

Specifies a module designed to perform the behavior of some component.

Routing Model

Entity that verifies the routing policy over the incoming message and, then, pass on the operative content to an essential model for processing.

Network Model

Routing scenario that structures the overall routing process as a composition of routing models.

RDEVS Models

Essential Model

Specifies a module designed to perform the behavior of some component.

(Component) DOMAIN BEHAVIOR

Routing Model

Entity that verifies the routing policy over the incoming message and, then, pass on the operative content to an essential model for processing.

Network Model

Routing scenario that structures the overall routing process as a composition of routing models.

RDEVS Models

Essential Model

Specifies a module designed to perform the behavior of some component.

(Component) DOMAIN BEHAVIOR

Routing Model

Entity that verifies the routing policy over the incoming message and, then, pass on the operative content to an essential model for processing.

(Component → Node) ROUTING BEHAVIOR

Network Model

Routing scenario that structures the overall routing process as a composition of routing models.

RDEVS Models

Essential Model

Specifies a module designed to perform the behavior of some component.

(Component) DOMAIN BEHAVIOR

Routing Model

Entity that verifies the routing policy over the incoming message and, then, pass on the operative content to an essential model for processing.

(Component → Node) ROUTING BEHAVIOR

Network Model

Routing scenario that structures the overall routing process as a composition of routing models.

(Process) STRUCTURE

RDEVS Models

Essential Model ⟷ Atomic Model

(Component) DOMAIN BEHAVIOR

Routing Model

(Component → Node) ROUTING BEHAVIOR

Network Model

(Process) STRUCTURE

RDEVS Models

Essential Model ⟷ Atomic Model

(Component) DOMAIN BEHAVIOR

Routing Model

- Essential Model
- Routing Policy

(Component → Node) ROUTING BEHAVIOR

Network Model

(Process) STRUCTURE

RDEVS Routing Model

Formal Definition

where

$\omega = (u, W, \delta_r) \equiv$ routing policy,

$E = < X_E, S_E, Y_E, \delta_{int,E}, \delta_{ext,E}, \lambda_E, \tau_E > \equiv$ RDEVS essential model embedded in R,

$M = <X_M, S_M, Y_M, \delta_{int,M}, \delta_{ext,M}, \lambda_M, \tau_M> \equiv$ DEVS atomic model that describes the behavior of the node to be executed during the routing process simulation.

RDEVS Routing Model

Formal Definition

where

$\omega = (u, W, \delta_r) \equiv$ routing policy,

$E = < X_E, S_E, Y_E, \delta_{int,E}, \delta_{ext,E}, \lambda_E, \tau_E > \equiv$ RDEVS essential model embedded in R,

$M = <X_M, S_M, Y_M, \delta_{int,M}, \delta_{ext,M}, \lambda_M, \tau_M> \equiv$ DEVS atomic model that describes the behavior of the node to be executed during the routing process simulation.

RDEVS Routing Model

Formal Definition

where

$\omega = (u, W, \delta_r) \equiv$ routing policy,

$E = < X_E, S_E, Y_E, \delta_{int,E}, \delta_{ext,E}, \lambda_E, \tau_E > \equiv$ RDEVS essential model embedded in R,

$M = <X_M, S_M, Y_M, \delta_{int,M}, \delta_{ext,M}, \lambda_M, \tau_M> \equiv$ DEVS atomic model that describes the behavior of the node to be executed during the routing process simulation.

RDEVS Routing Model

Formal Definition

$X_M = \{ (x, h, T_{IN}) \mid$ [redacted] $h \in N_0,\ T_{IN} = \{ t_1, t_2, \ldots, t_k \mid t_1, t_2, \ldots, t_k \in N_0 \} \} \equiv$

set of input events identified inside the routing process,

$Y_M = \{ (y, u, T_{OUT})$ [redacted] $u \in N_0,\ T_{OUT} = \{ t_1, t_2, \ldots, t_k \mid t_1, t_2, \ldots, t_k \in N_0 \} \} \equiv$

set of output events identified inside the routing process, with

$S_M =$ [redacted] $\equiv$ set of sequential states,

$\delta_{int,M}: S_M \rightarrow S_M =$ [redacted] $\equiv$ internal transition function,

$\delta_{ext,M}: Q_M \times X_M \rightarrow S_M \equiv$ external transition function, defined as

$$\delta_{ext,M}(s,e,x') = \begin{cases} \text{[redacted]} & \text{if } \bigcirc \text{ with } x' = (x, h, T) \\ s & \text{otherwise} \end{cases}$$

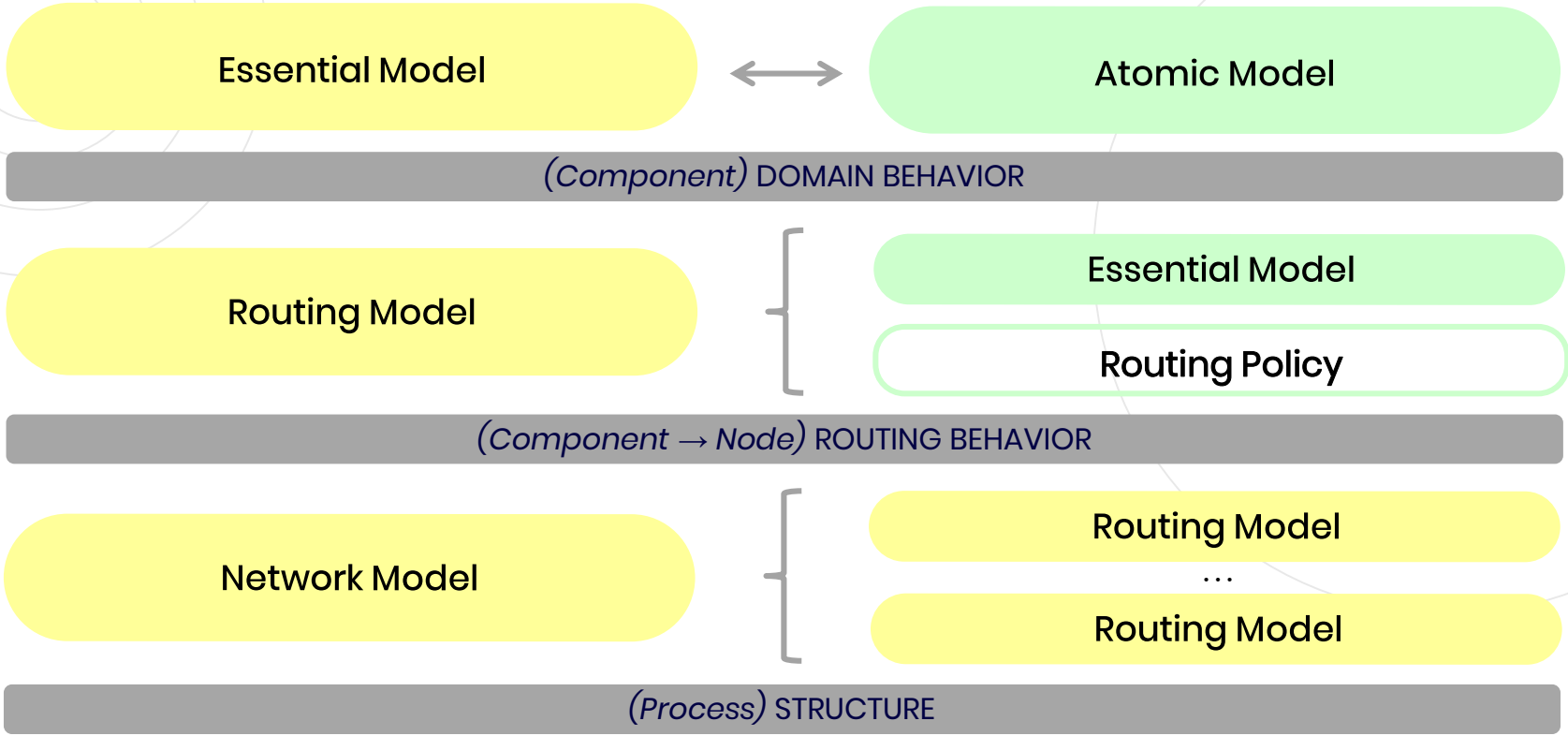
$$\bigcirc\ (u \in T_{IN} \wedge h \in W) \vee (h = 0 \wedge u \in T_{IN}) \vee (u = 0 \wedge W = \emptyset)$$

$\lambda_M: S_M \rightarrow Y_M \cup \emptyset \equiv$ output function, defined as

$$\lambda_M(s) = \text{[redacted]}\ (u, \delta_r(s))$$

$\tau_M: S_M \rightarrow R^+_{0,\infty} \equiv$ time advance function.

RDEVS Models



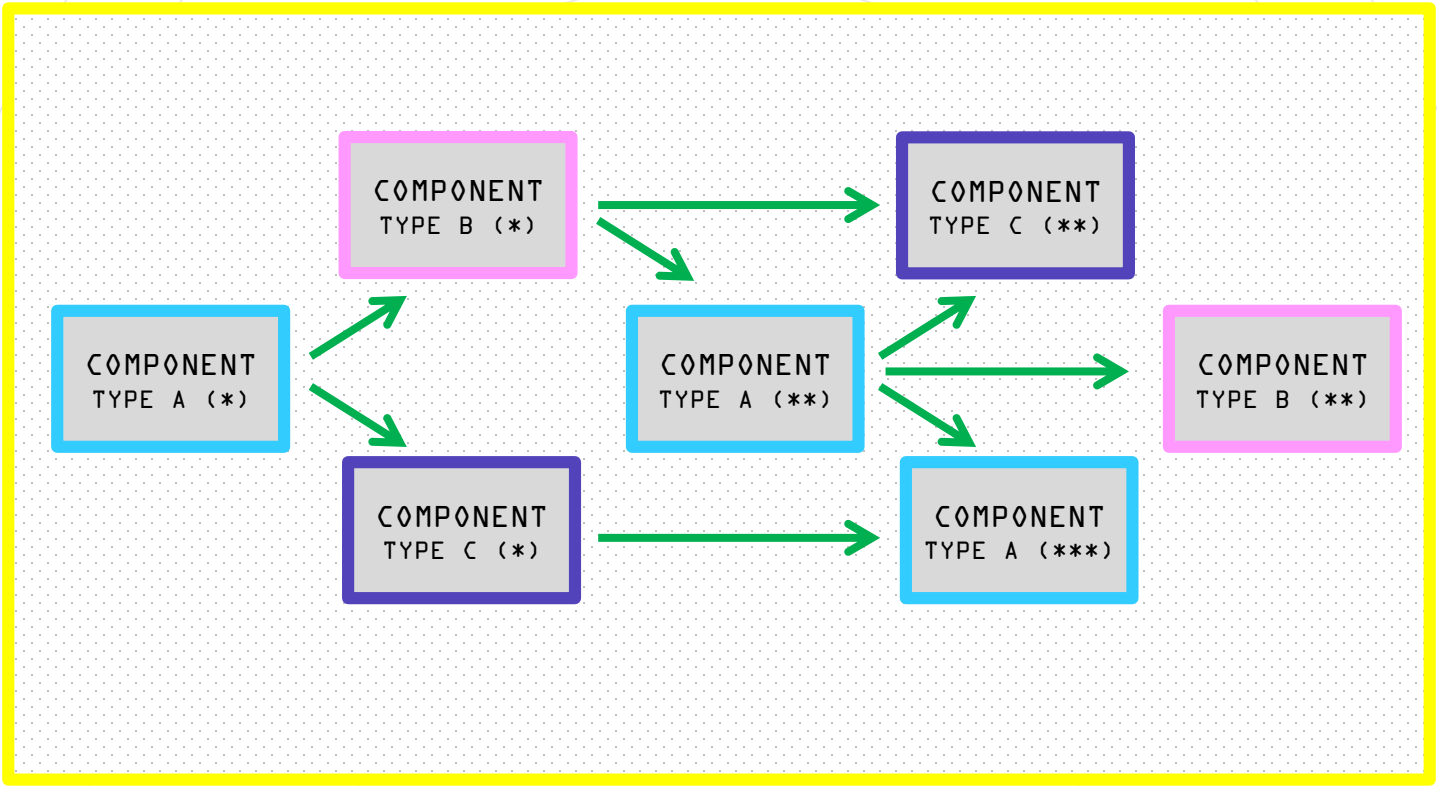
Closure Under Coupling

A system formalism is closed under coupling if the resultant of any network of systems specified in the formalism is itself a system in the formalism.

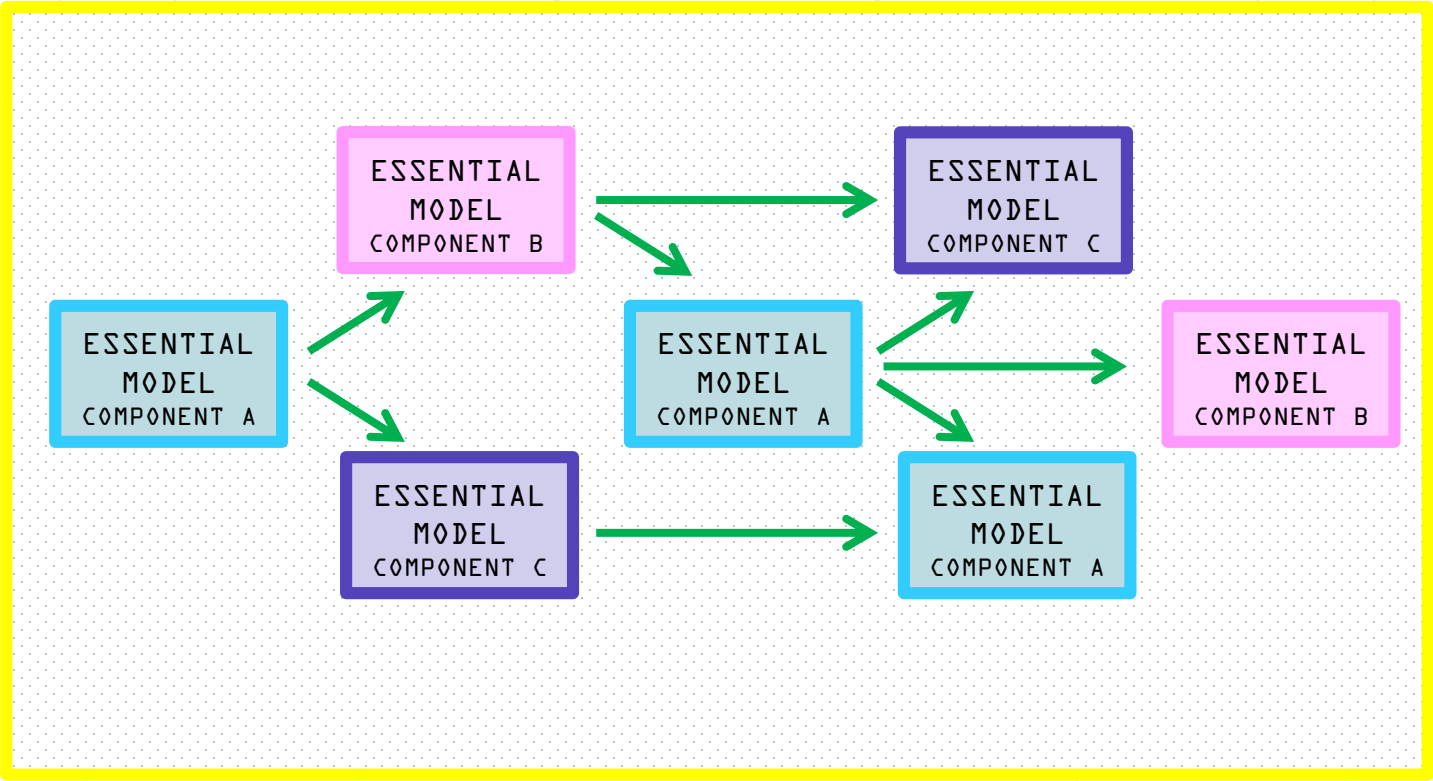
To prove that RDEVS formalism is closed under coupling, two new models were obtained:

- the routing model that is *behavioral equivalent* to the network model, and
- the essential model that is *behavioral equivalent* to the routing model.

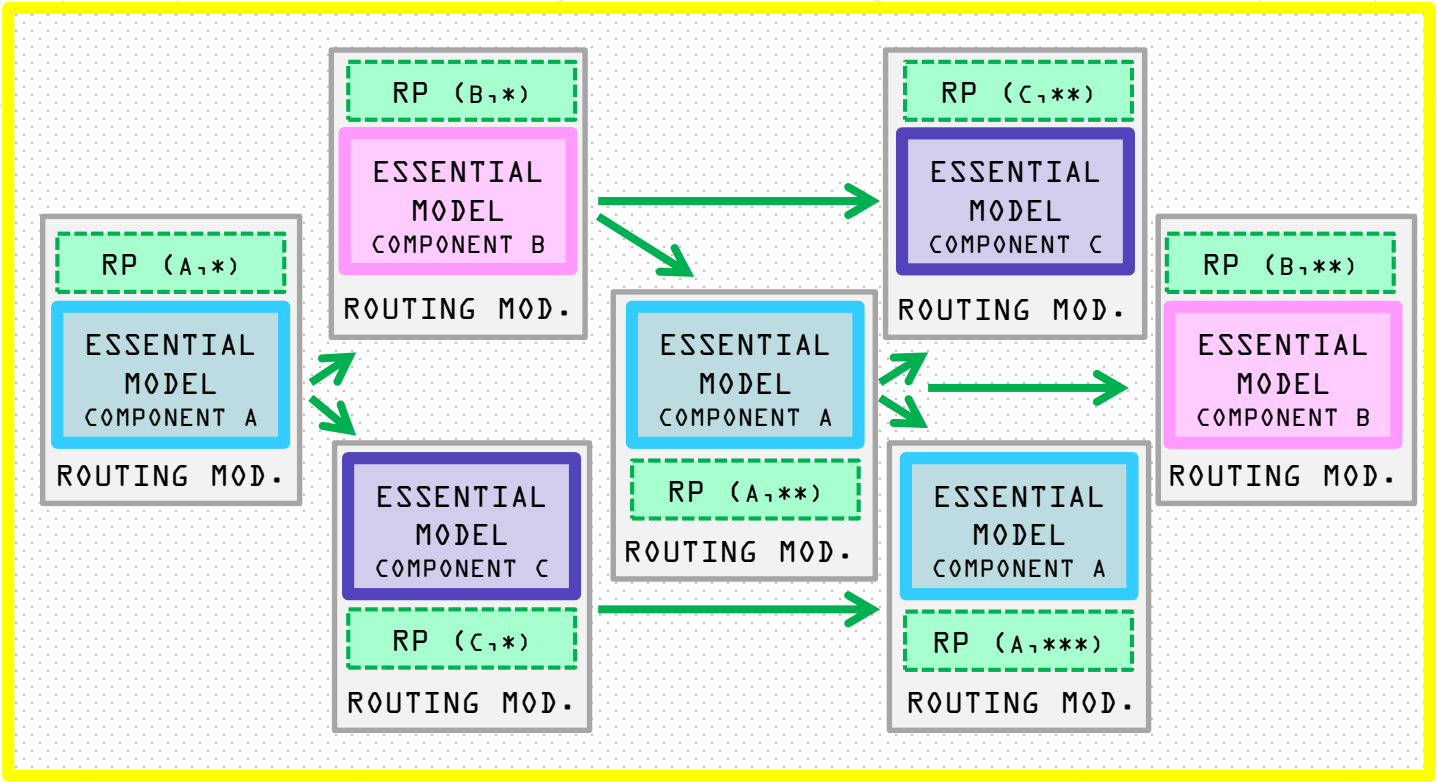
The **hierarchical construction** stays within the **formalism**.



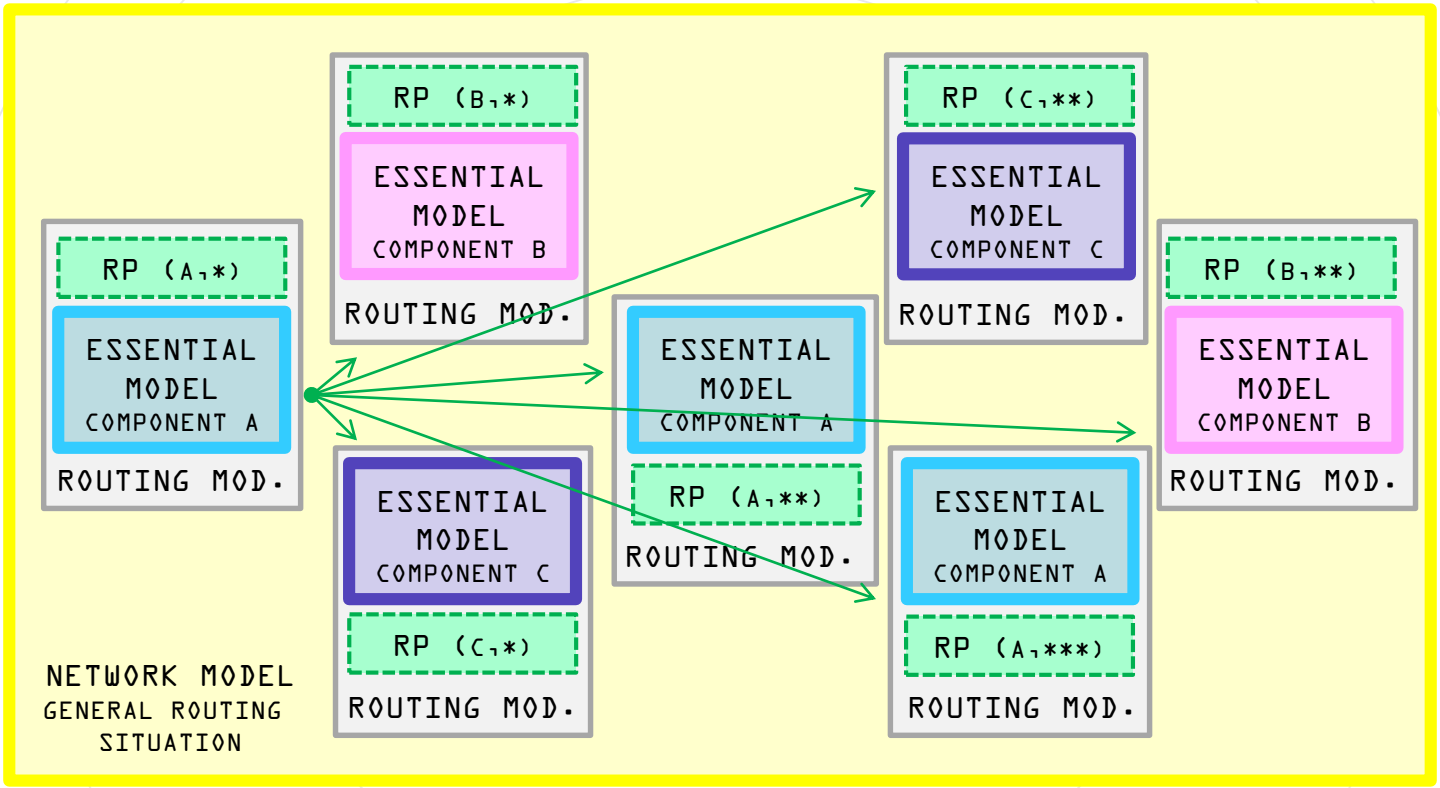
Routing Problem (RDEVS Solution)



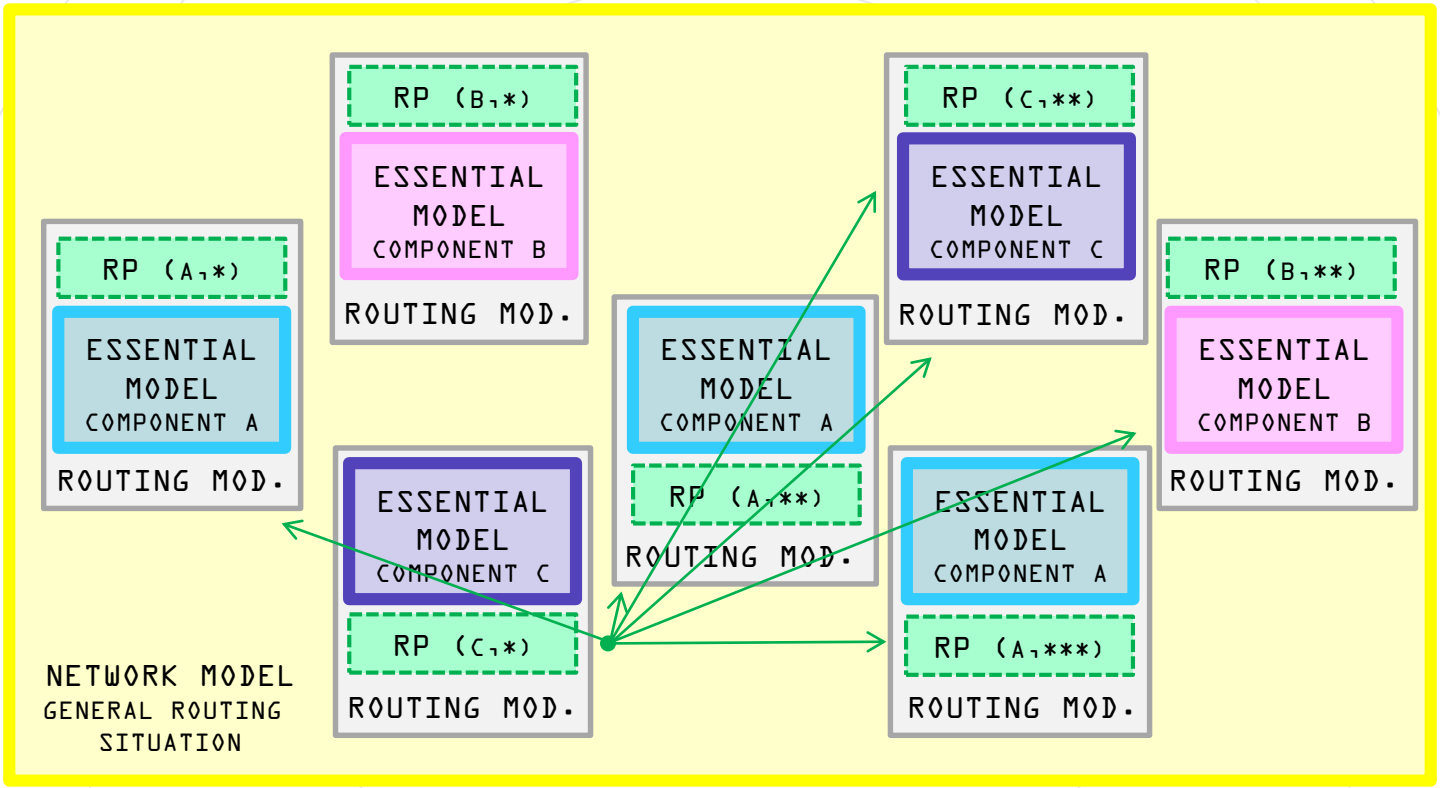
Routing Problem (RDEVS Solution)



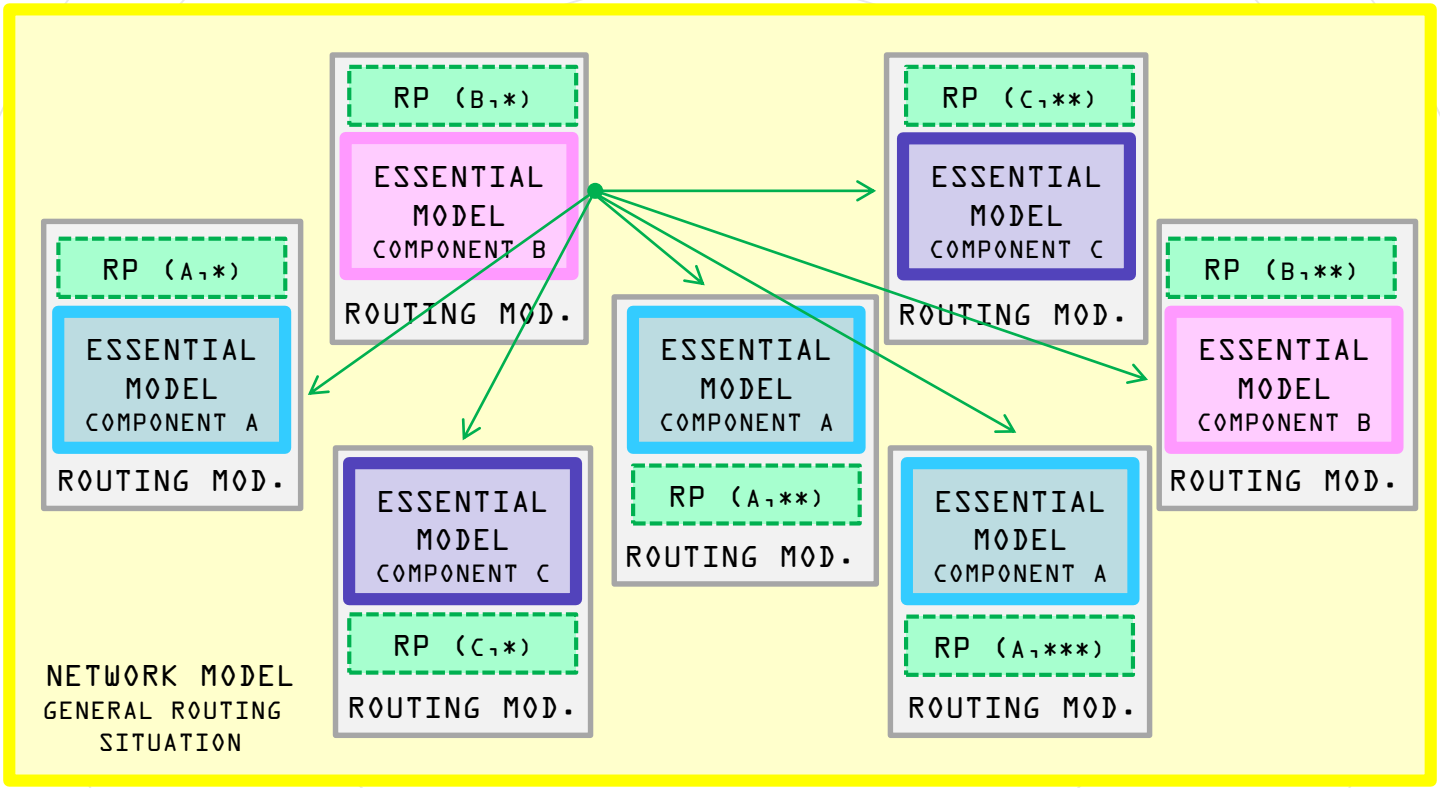
Routing Problem (RDEVS Solution)



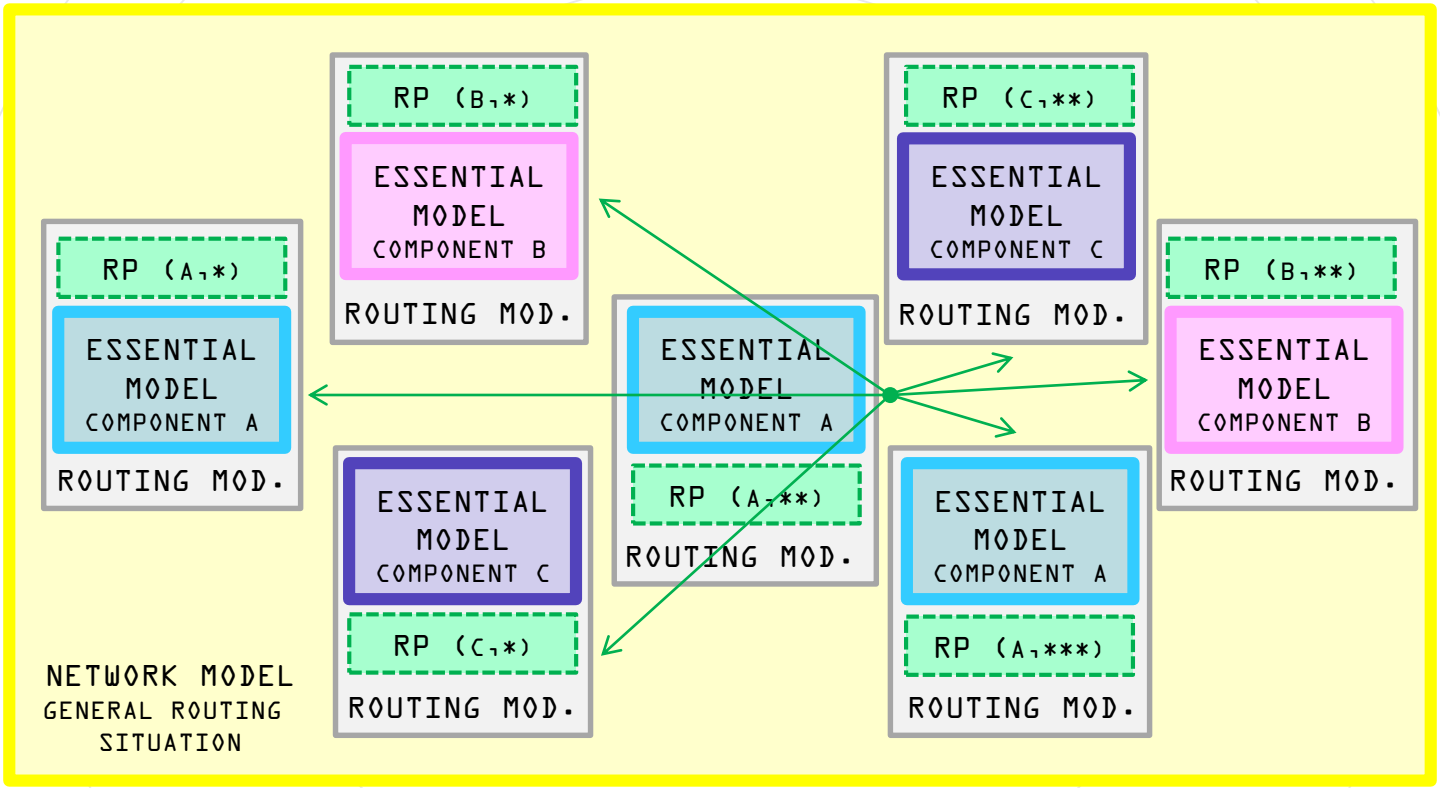
Routing Problem (RDEVS Solution)



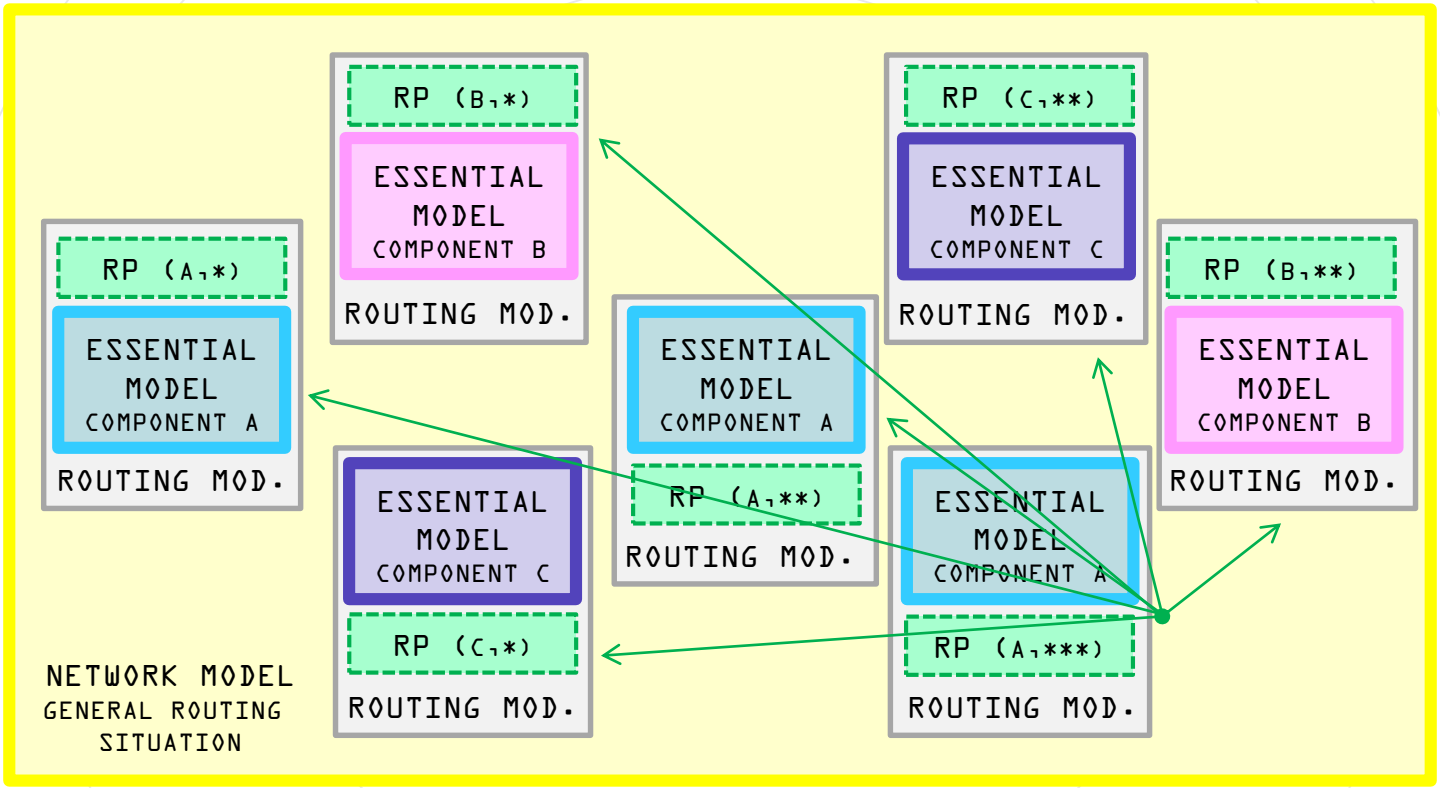
Routing Problem (RDEVS Solution)



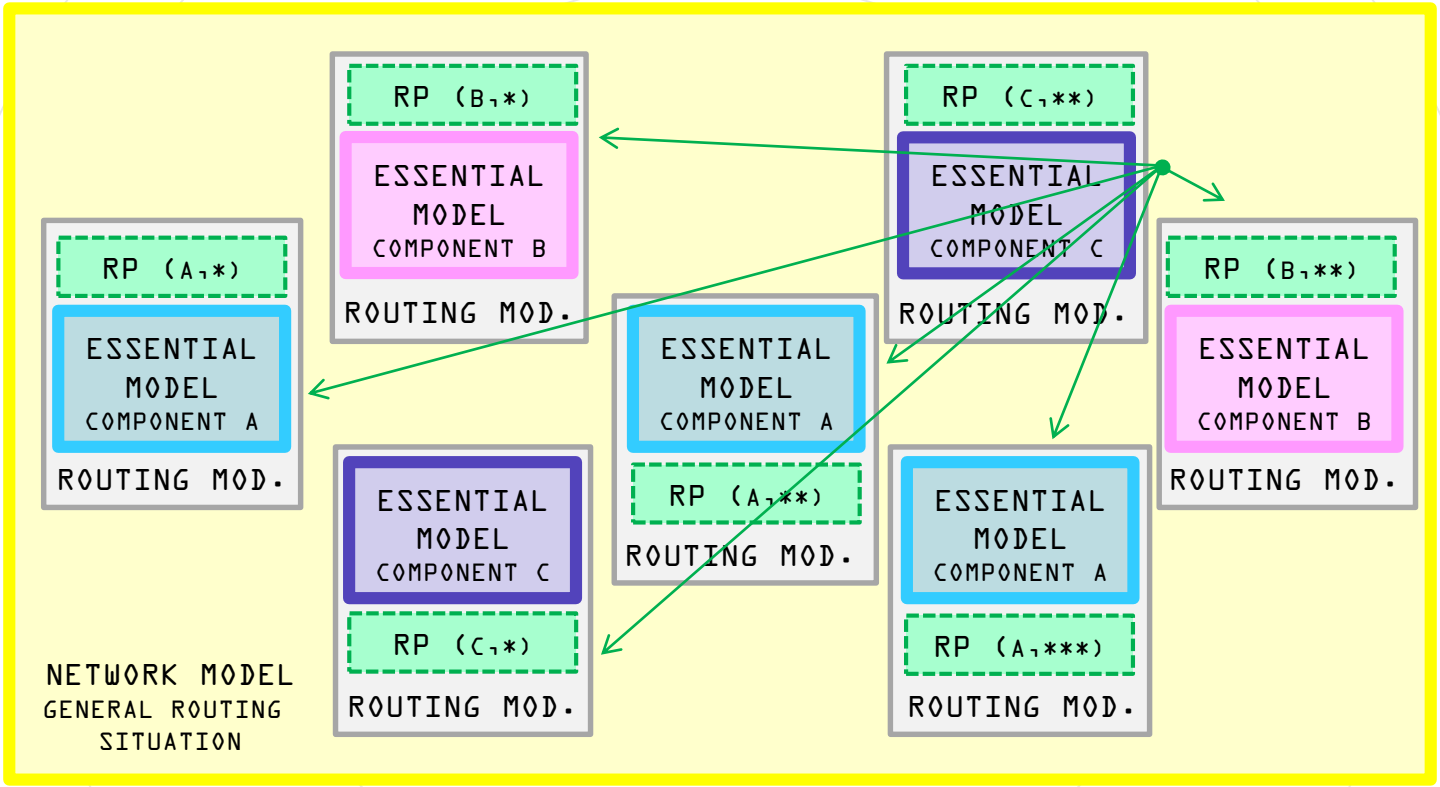
Routing Problem (RDEVS Solution)



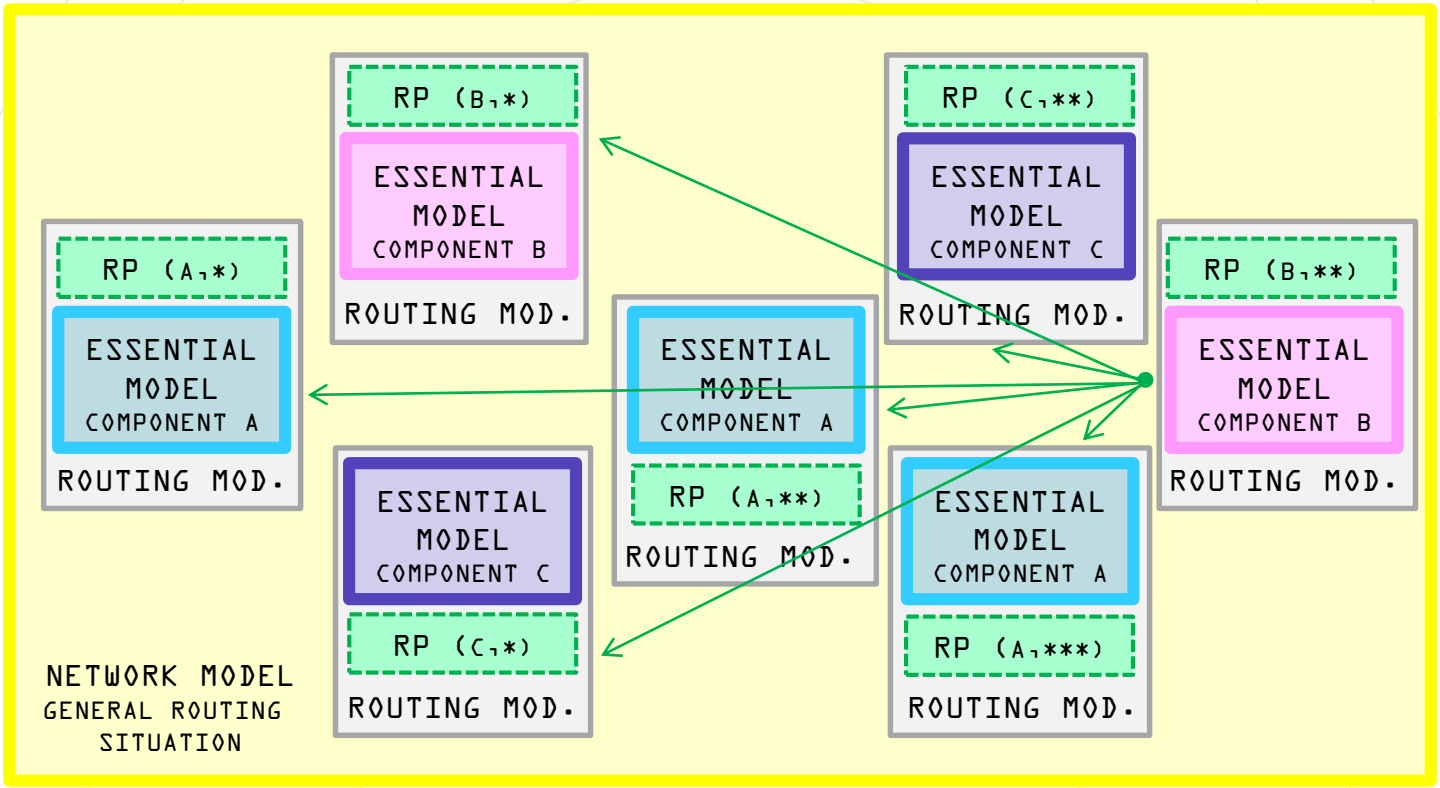
Routing Problem (RDEVS Solution)



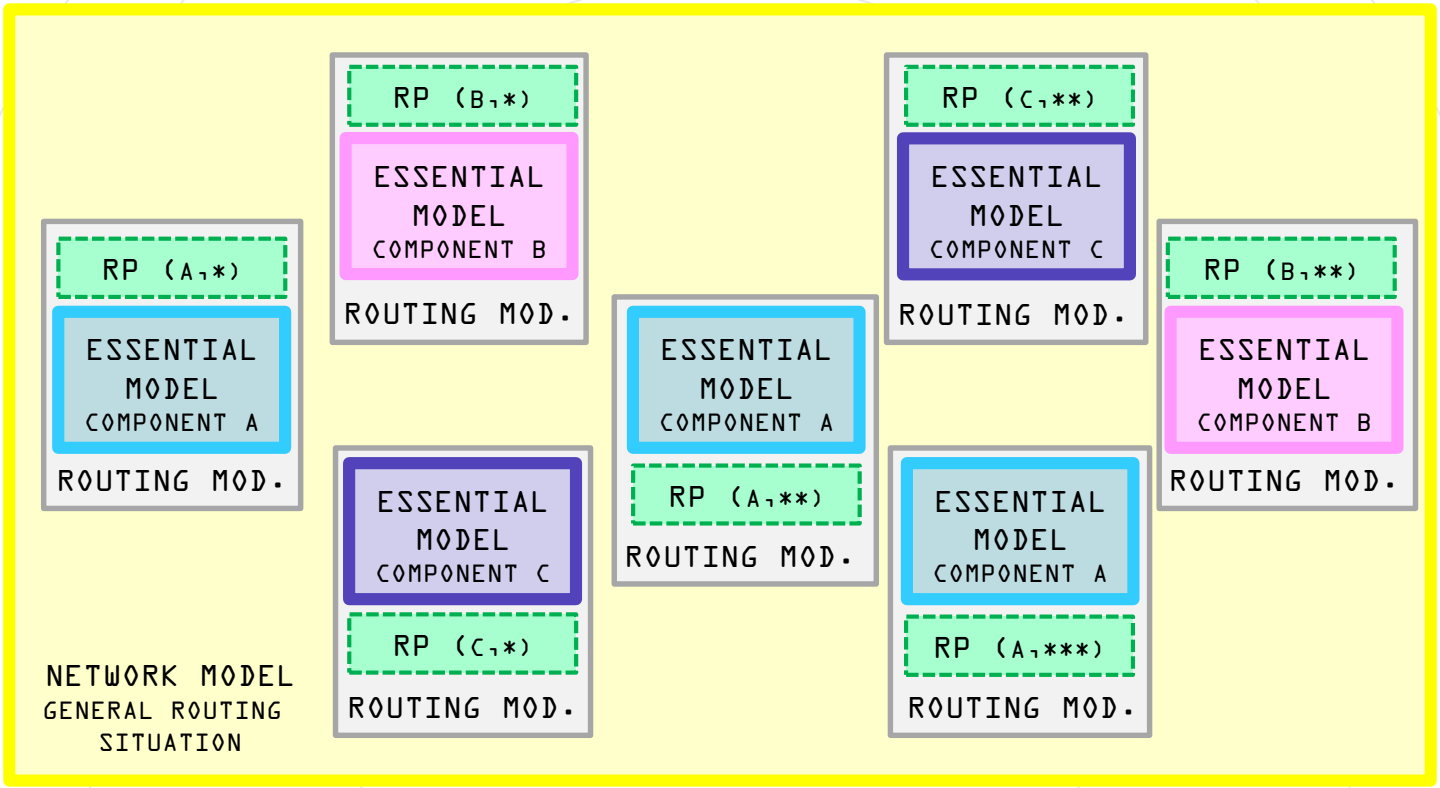
Routing Problem (RDEVS Solution)



Routing Problem (RDEVS Solution)



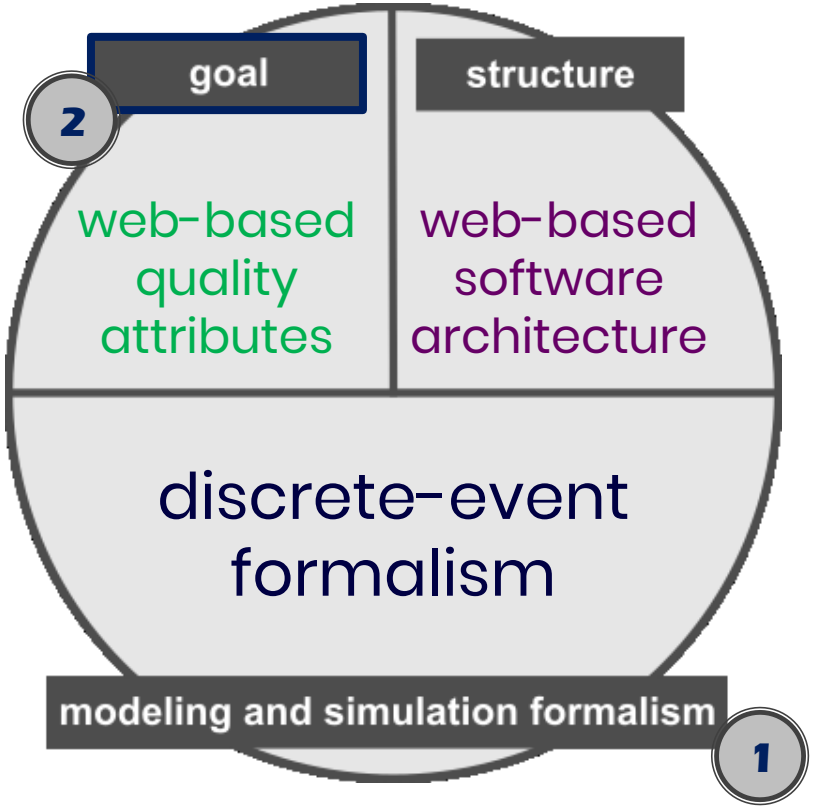
Routing Problem (RDEVS Solution)



Routing Problem (RDEVS Solution)

Thesis Contributions

Composition Level



2

Web-based Applications

Quality Attributes

Quality Scheme for Web Applications

Instance of the QSO Ontology

A Quality Scheme was defined to study the set of software attributes commonly measured in web-based applications.

Web Quality Scheme

{ software attribute,
software metric,
quality property }

Quality Properties Evaluation

Following the software product **quality model** of ISO/IEC 25010.

Quality Scheme for Web Applications

Instance of the QSO Ontology

A Quality Scheme was defined to study the set of software attributes commonly measured in web-based applications.

Quality Measures

Data required to calculate the available set of metrics in a specific development phase.

Web Quality Scheme

{ software attribute, software metric, quality property }

An information extraction process was applied with aims to obtain the set of variables to be measured during the simulation.

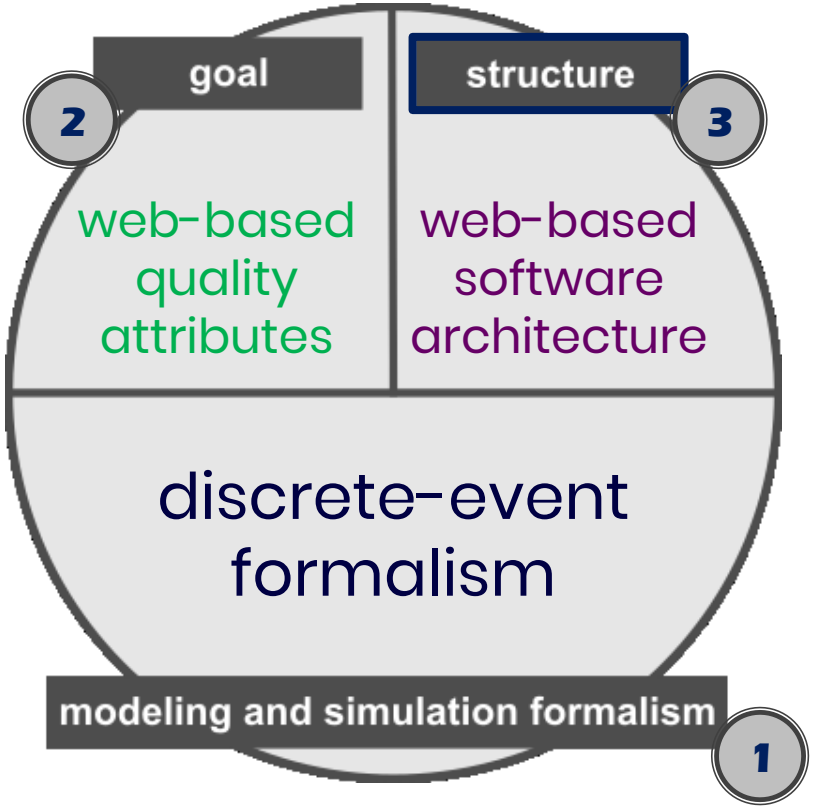
Simulation Goals

Web Quality Scheme → Direct Metrics

VARIABLE	DESCRIPTION	UNIT	WEB-SOFTWARE ATTRIBUTE
ET	Processing time for user requests.	time	Invocation time
TSIT	Total time used for processing a user request.	time	
TR	Number of requests processed by the application.	requests	Service accuracy
IR	Number of incorrect responses to requests.	requests	
FT	Inactive time.	time	Robustness of service
TT	Operative time.	time	
FNF	Number of faults that are not failures.	faults	Service stability
TF	Number of faults.	faults	
RF	Number of failures solved.	failures	

Thesis Contributions

Composition Level



3

Web-based Applications

Software Architecture

“ Quality attributes should be achieved using well-known architectural patterns.

“

Quality attributes should be achieved using well-known architectural patterns.

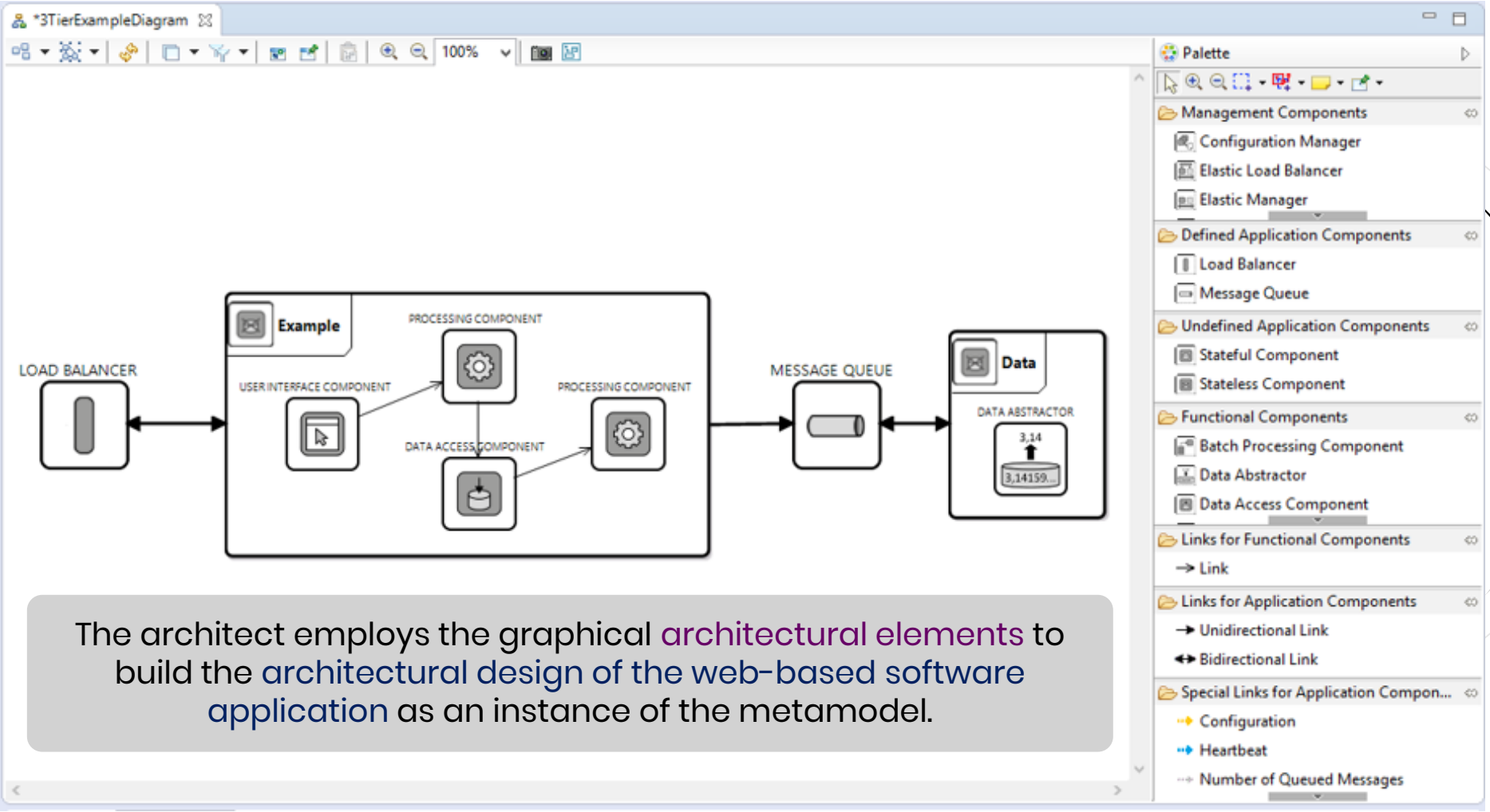
Web architectural patterns were studied with aim to identify the components commonly used in web architectures.

Fehling et al. (2014). *Cloud computing patterns: fundamentals to design, build, and manage cloud applications.*

Metamodel for Building Web Architecture Designs

- These components were classified in an UML metamodel.
- The UML metamodel was completed with OCL constraints to ensure the accuracy of web-based architectures instantiated using the patterns.

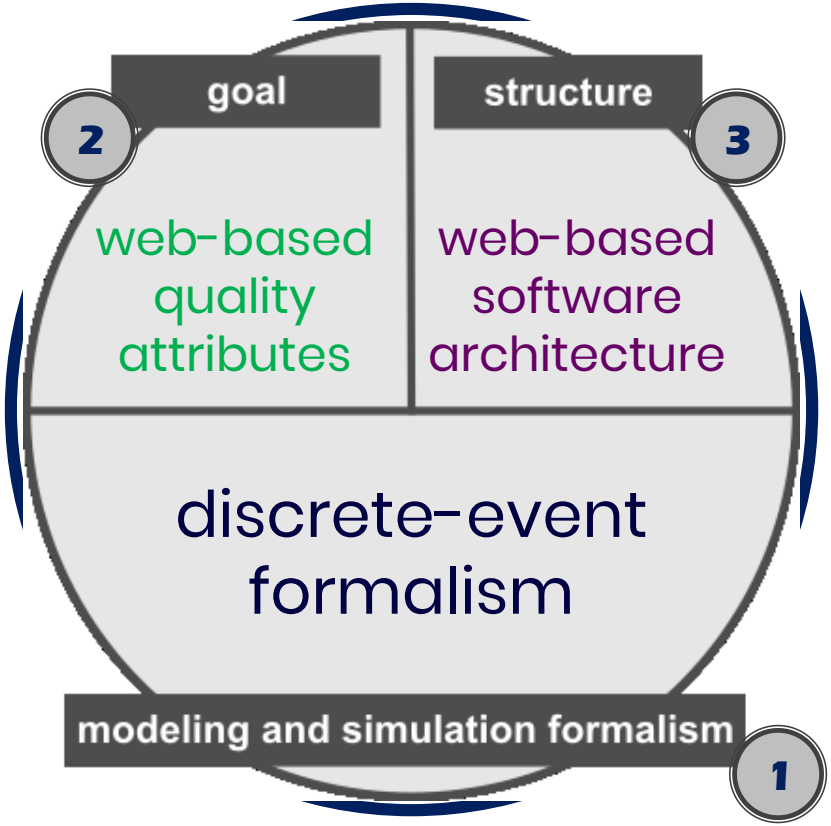
The metamodel was used as architectural language for the development of a software modeling tool.



The architect employs the graphical architectural elements to build the architectural design of the web-based software application as an instance of the metamodel.

Thesis Contributions

Composition Level



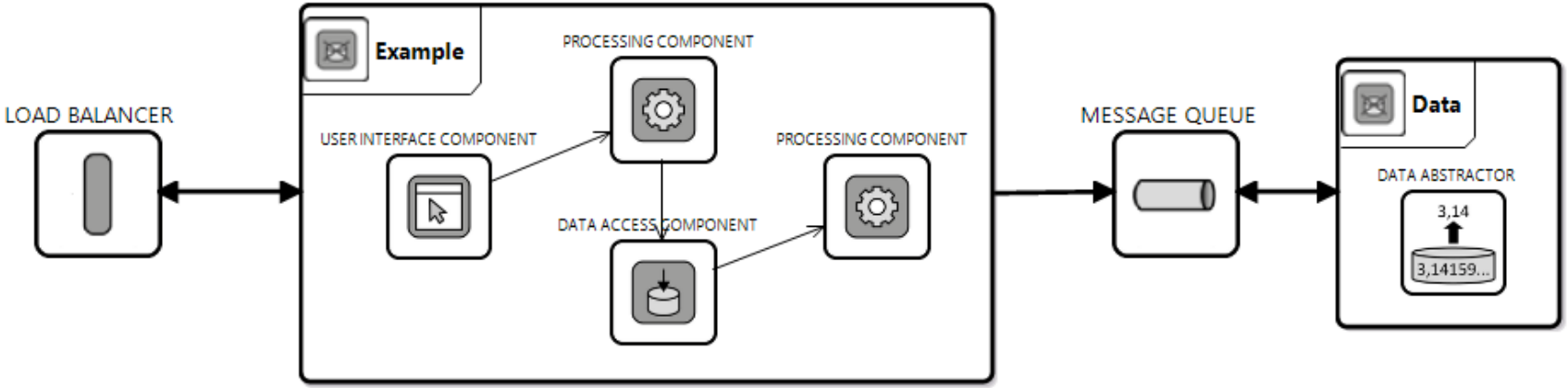
4

Simulation Model

Web-based Software Architecture for Quality Estimation

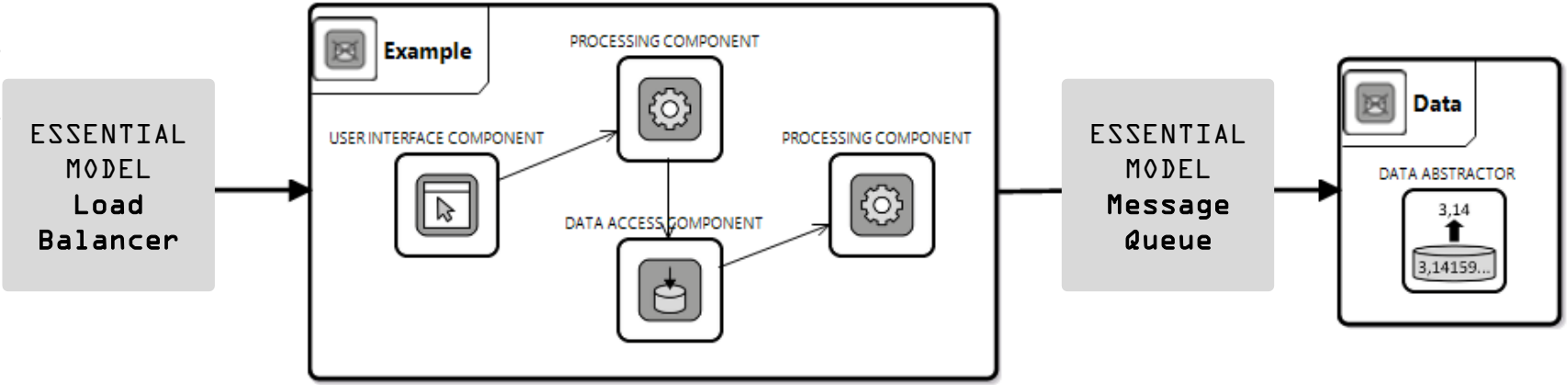
Web Application Architecture

Example: Web-based Software Architecture Design



Web Application Architecture

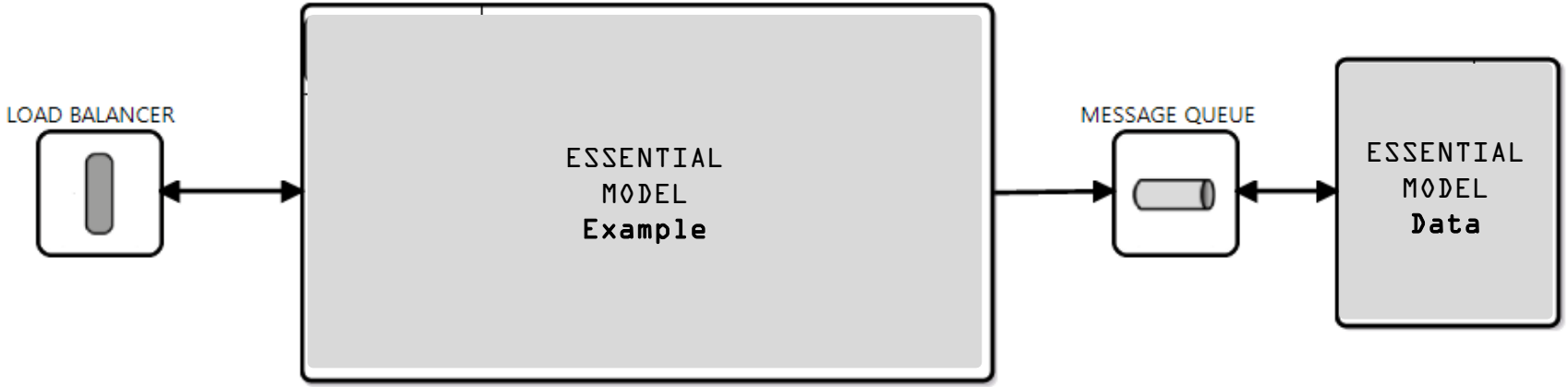
Example: Architecture Design to RDEVS Essential Model



Components frequently used in web-based software.

Web Application Architecture

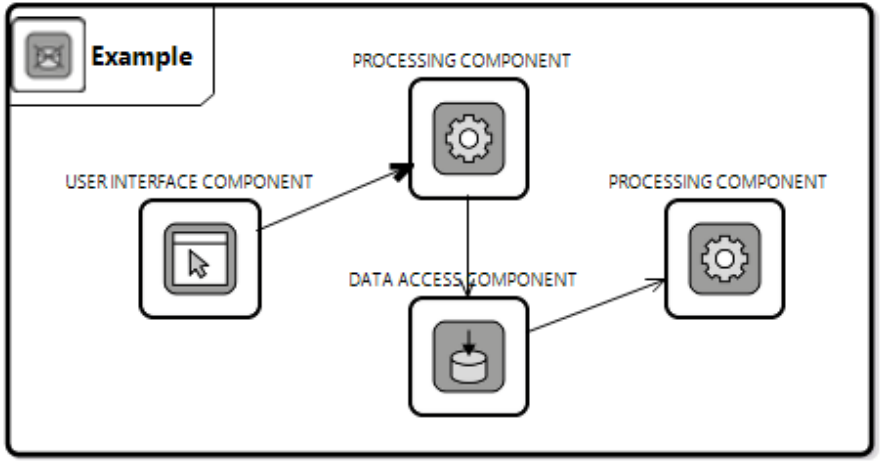
Example: Architecture Design to RDEVS Essential Model



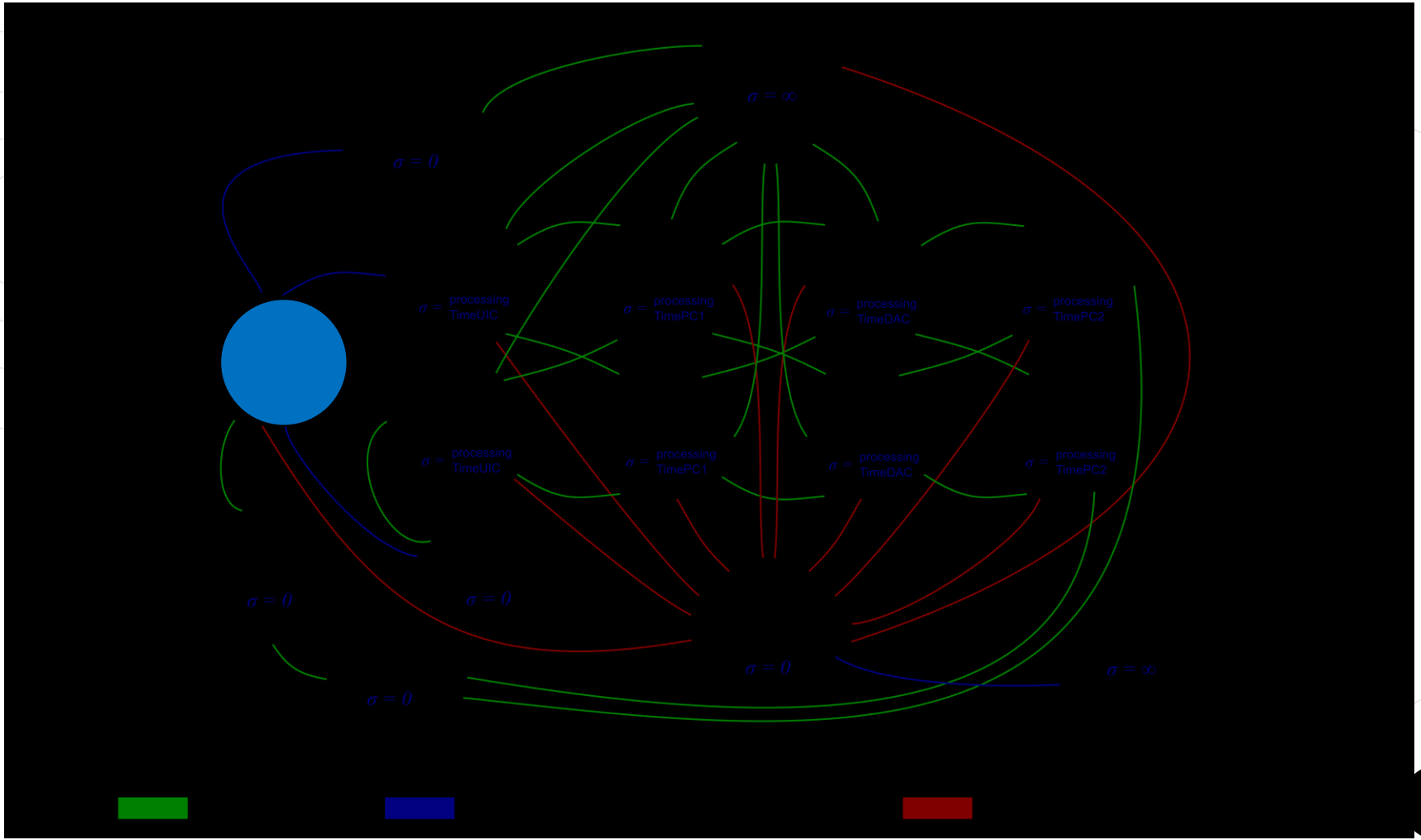
Domain-specific components.

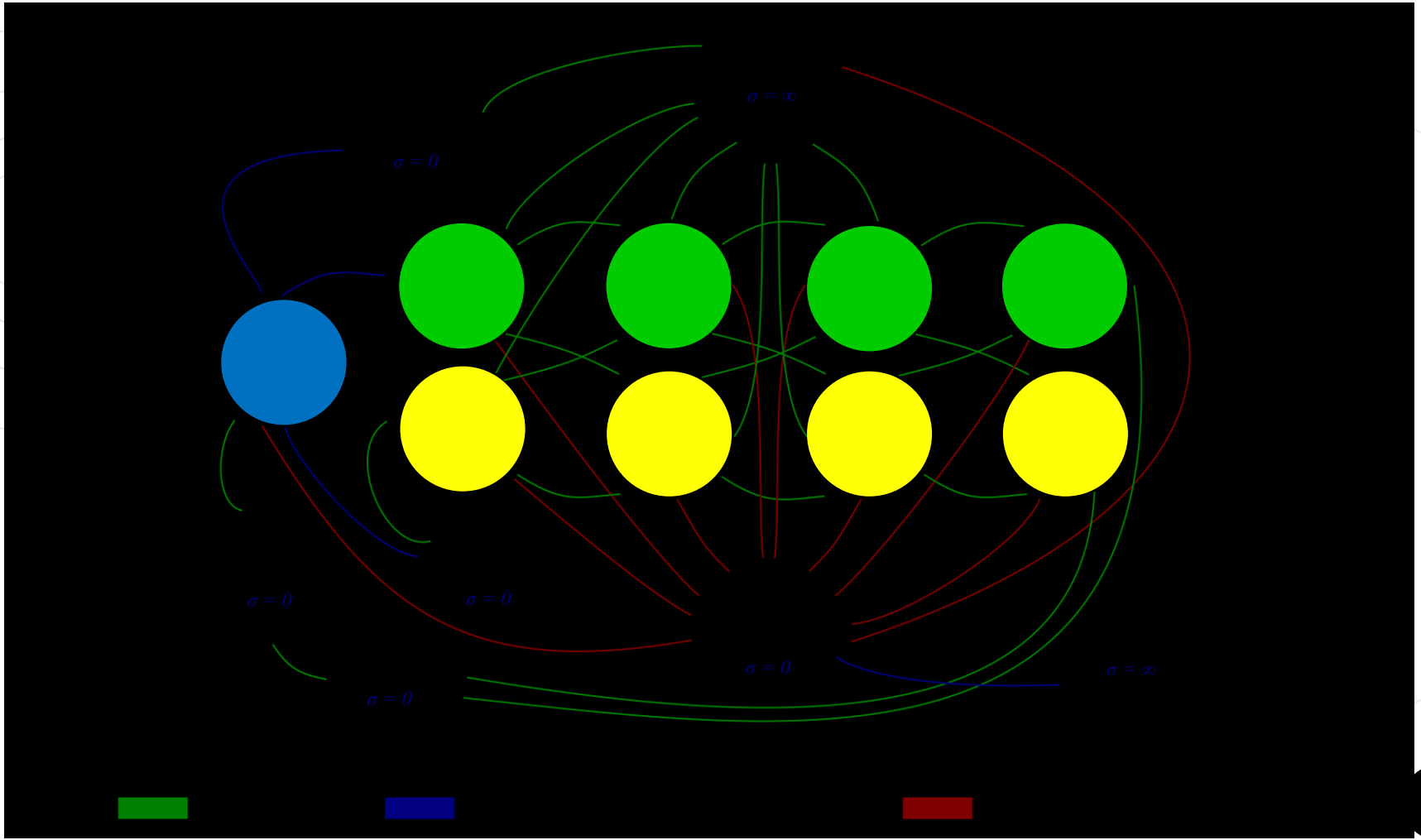
Web Application Architecture

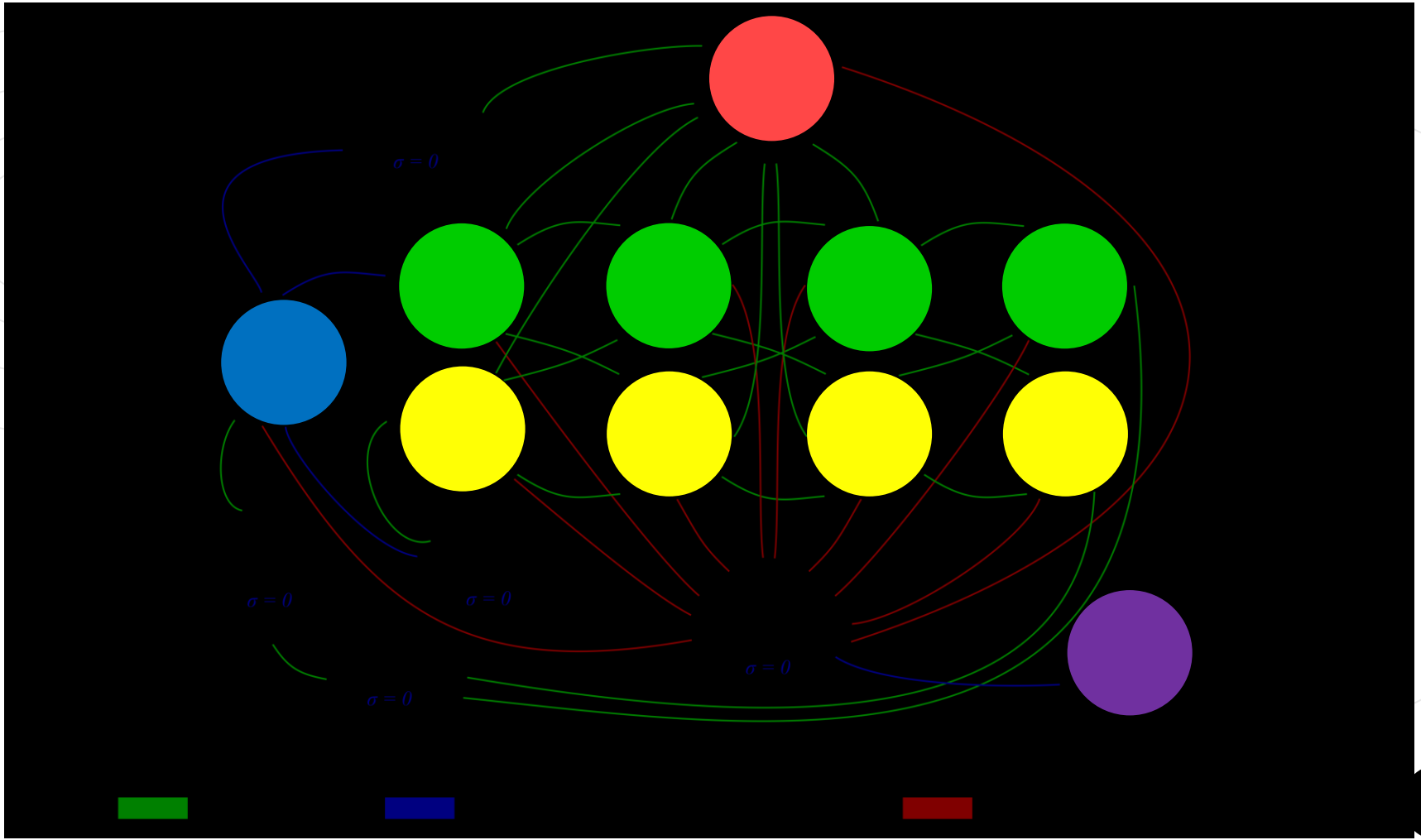
Example: RDEVS Essential Model “Example”



ESSENTIAL
MODEL
Example

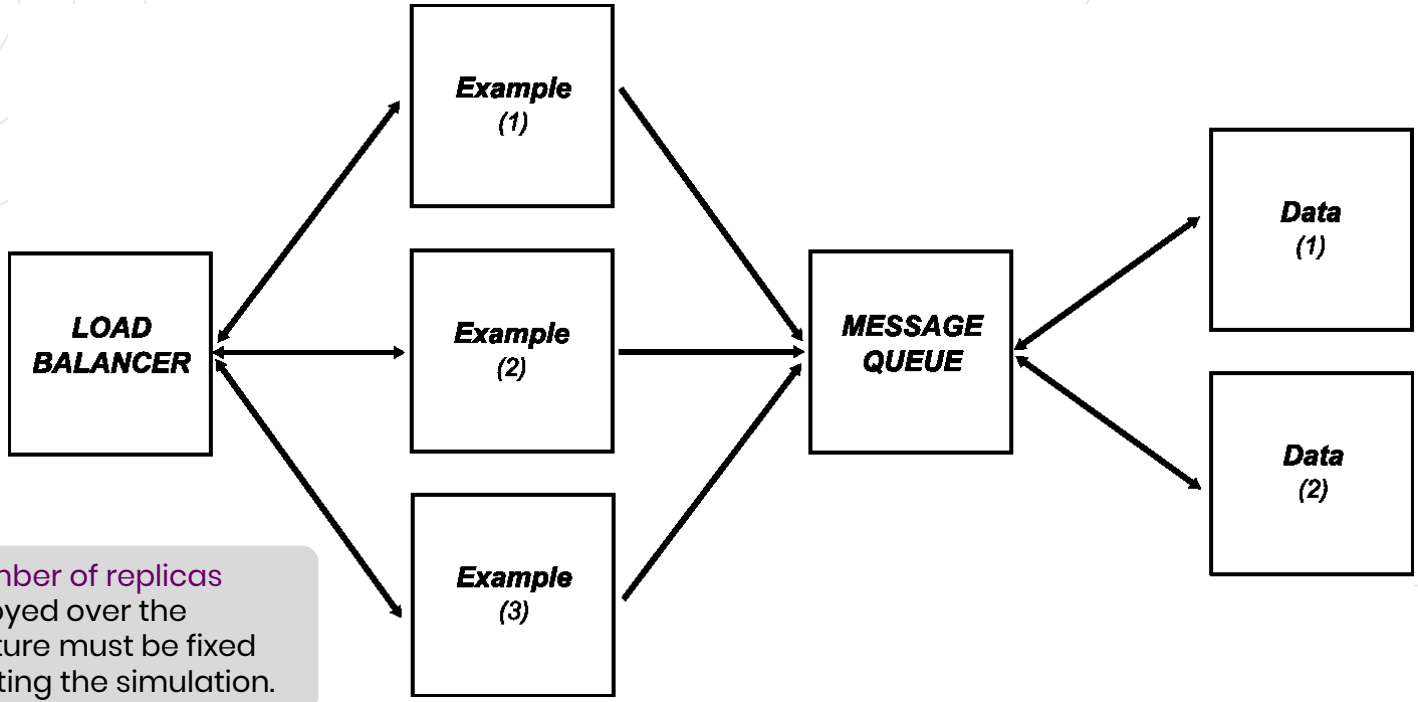

$\sigma = \infty$
$\sigma = 0$
$\sigma = \text{processing TimeUIC}$
$\sigma = \text{processing TimePC1}$
$\sigma = \text{processing TimeDAC}$
$\sigma = \text{processing TimePC2}$
$\sigma = \text{processing TimeUIC}$
$\sigma = \text{processing TimePC1}$
$\sigma = \text{processing TimeDAC}$
$\sigma = \text{processing TimePC2}$
$\sigma = 0$
$\sigma = 0$
$\sigma = 0$
$\sigma = 0$
$\sigma = \infty$


$\sigma = \infty$
$\sigma = 0$
$\sigma = 0$
$\sigma = 0$
$\sigma = 0$
$\sigma = 0$
$\sigma = \infty$


$\sigma = 0$
$\sigma = 0$
$\sigma = 0$
$\sigma = 0$
$\sigma = 0$

Web Application Architecture

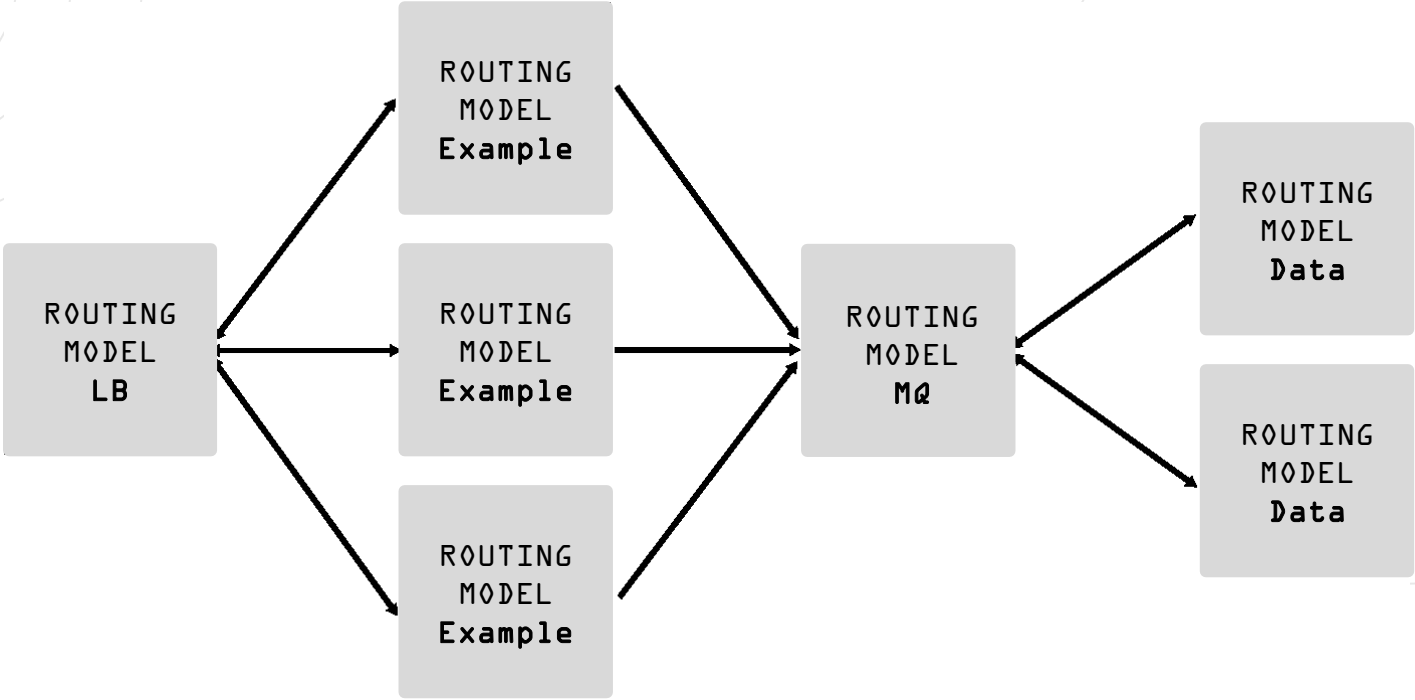
Example: Web-based Software Architecture Deployment



The number of replicas deployed over the infrastructure must be fixed prior executing the simulation.

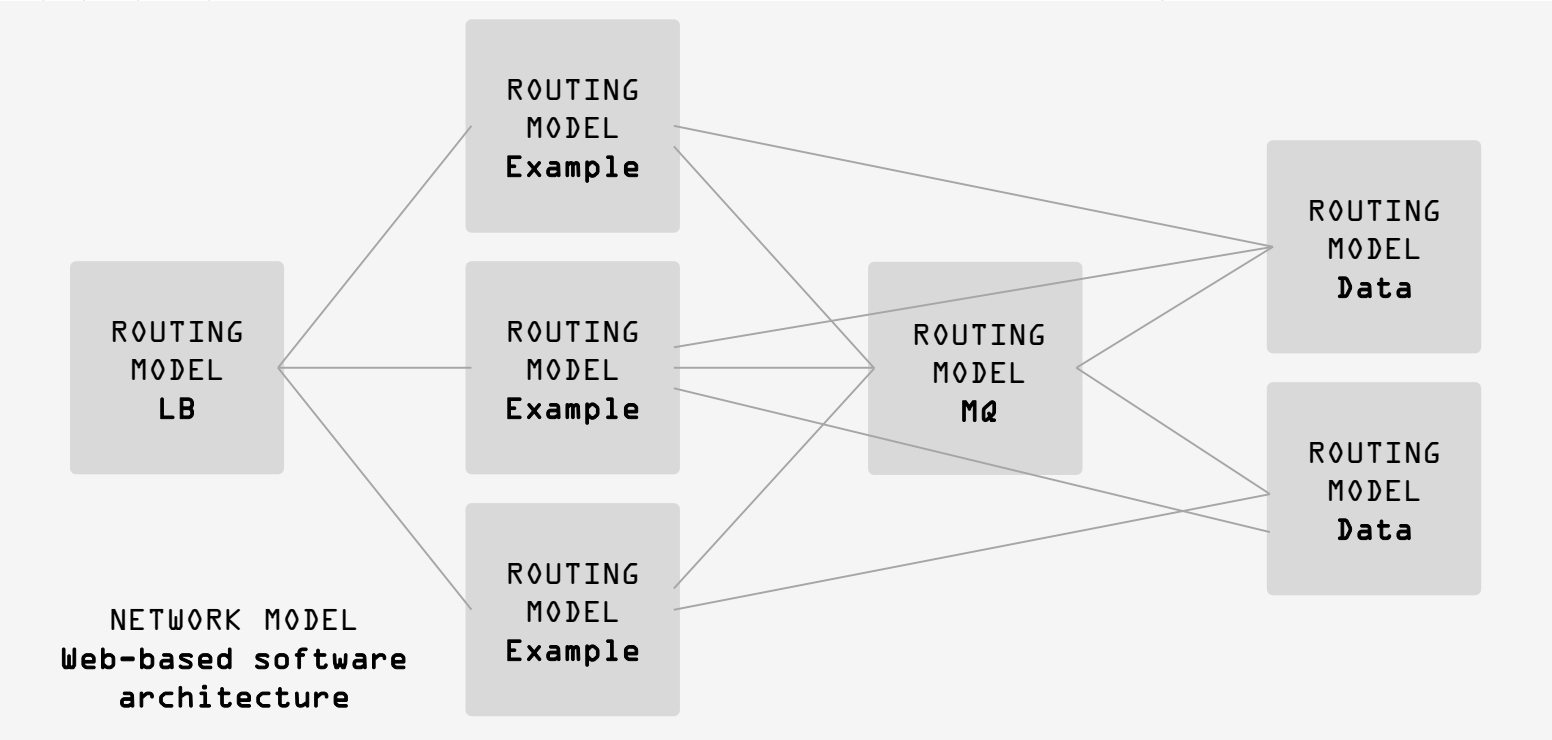
Web Application Architecture

Example: Architecture Deployment to RDEVS Network Model



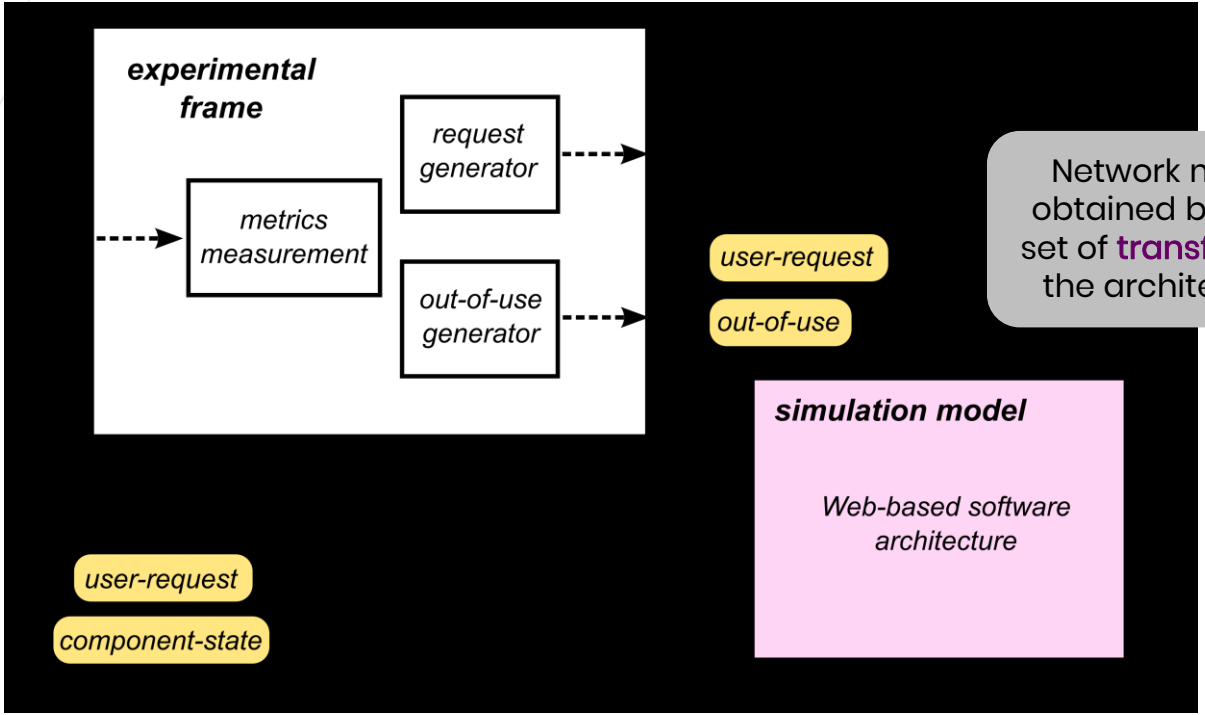
Web Application Architecture

Example: RDEVS Network Model



Simulation Model

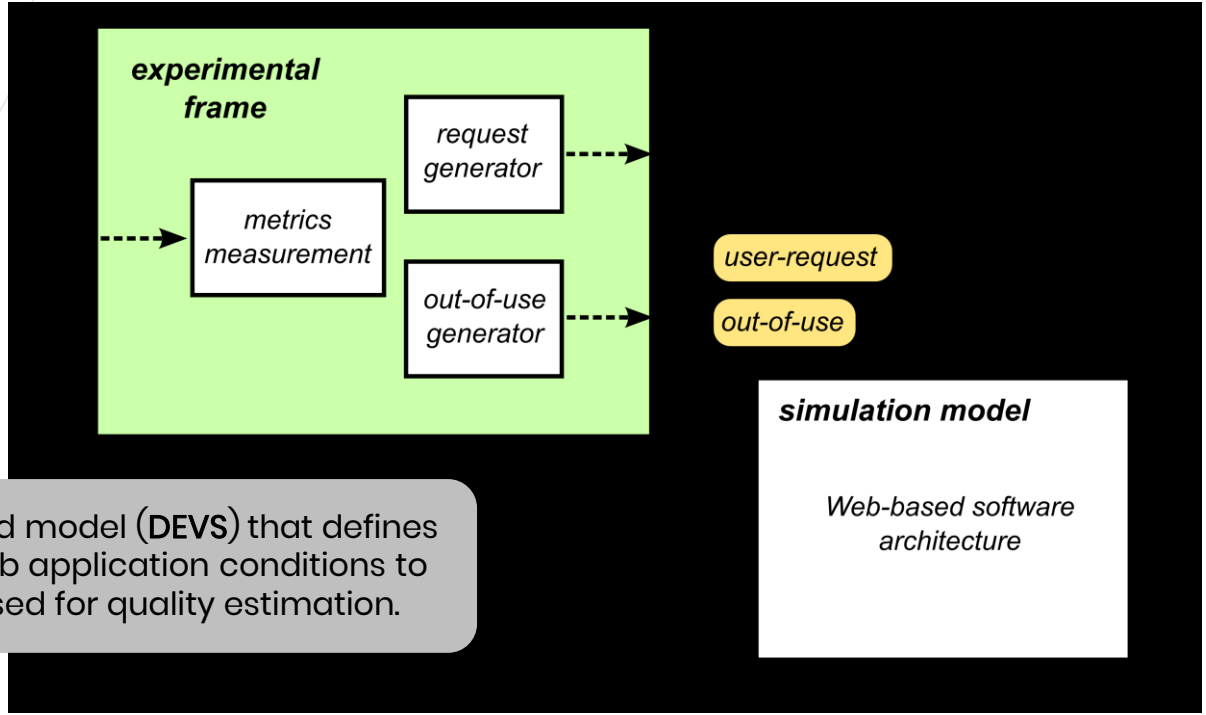
Final Structure



Network model (**RDEVS**) obtained by performing a set of **transformations** over the architectural design.

Simulation Model

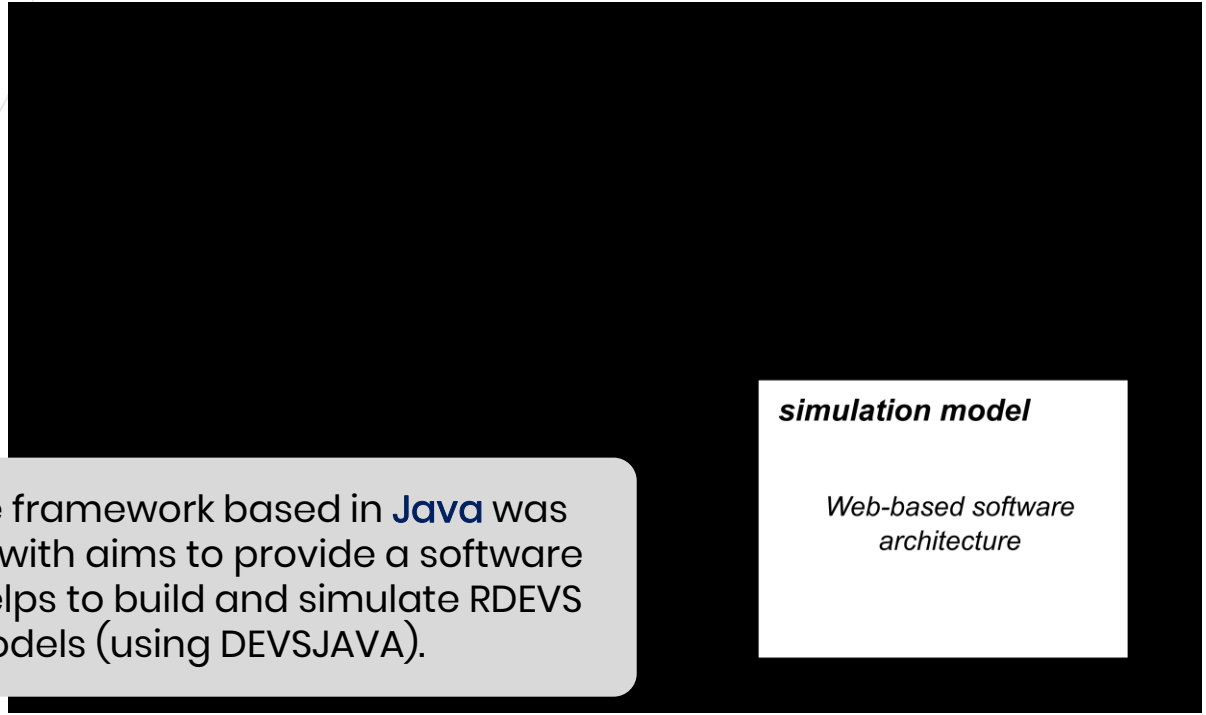
Final Structure



Coupled model (DEVS) that defines the web application conditions to be used for quality estimation.

Simulation Model

Final Structure



A software framework based in Java was developed with aims to provide a software tool that helps to build and simulate RDEVS models (using DEVSJAVA).

5

Evaluation of the M&S Contribution

…to the Software Engineering Field

Web-based Architectures

Two case study were studied using the simulation model obtained from the architectural design.

Number of Replicas	Two Tiers			Three Tiers		
	Network model	*Routing model*	*Essential model*	*Network model*	*Routing model*	*Essential model*
5						
10	1	11	2	1	33	6
20						
25	1	26	2	1	78	6
50						

Web-based Architectures

The simulation results obtained in both cases allow to validate the thesis contributions from the software engineering perspective.

Software architects do not need to be aware of discrete-event formalisms for using the simulation models for quality estimation.

The application of the RDEVS formalism as modeling specification of software architectures was successful.

6

Conclusions

and Future Research

Contributions

To the Software Engineering Field…

- The simulation models obtained from the architecture design provide a new technique for quality evaluation in early stages of development.

To the M&S Field…

- The RDEVS formalism as a standard solution for building simulation models that structure routing processes.

The proposal disadvantages are given by the assumptions made at the beginning of the simulation modeling tasks.

Future Work

Quality evaluation of self-adaptive architectures using an improved version of the actual simulation models combined with other discrete-event approaches that take advantage of their dynamic structure.

María Julia Blas

mariajuliablas@santafe-conicet.gov.ar

INGAR CONICET UTN