Analytical and Hierarchical Models for Availability Evaluation of Edge-Fog-Cloud Continuum Applications

Paulo Pereira
prps@cin.ufpe.br

Advisor: Dr. Paulo Maciel
prmm@cin.ufpe.br

Co-advisor: Dr. Jean Araújo
jean.teixeira@ufape.edu.br

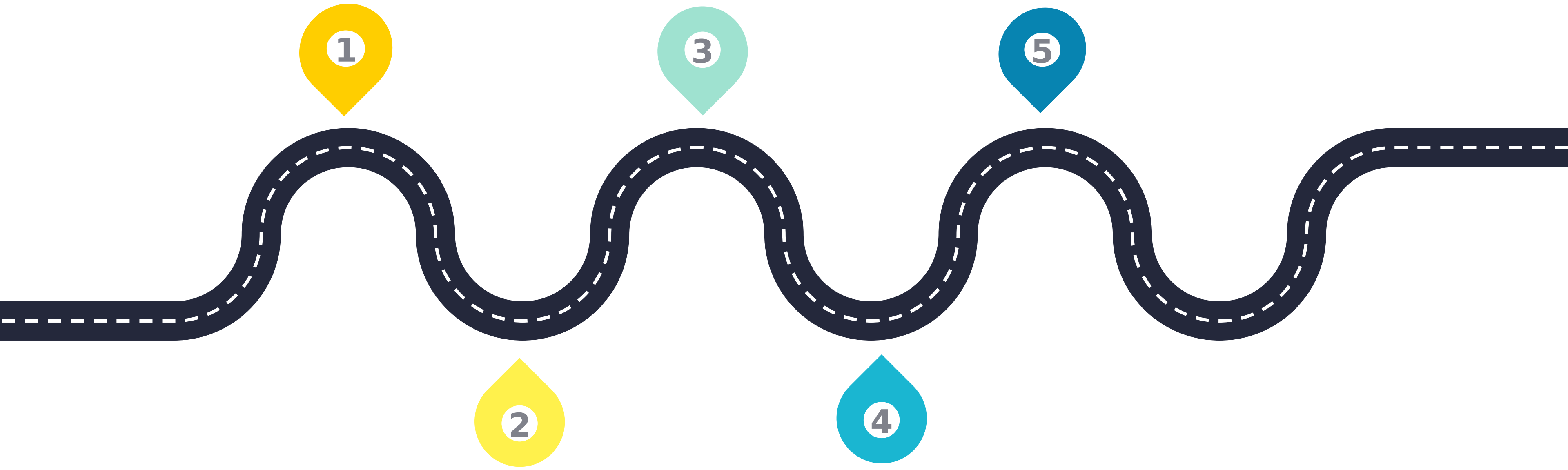
Centro de Informática UFPE
Universidade Federal de Pernambuco

Agenda

Introduction

Proposal

Conclusion



Backgroun
d

Case
Studies

Introduction

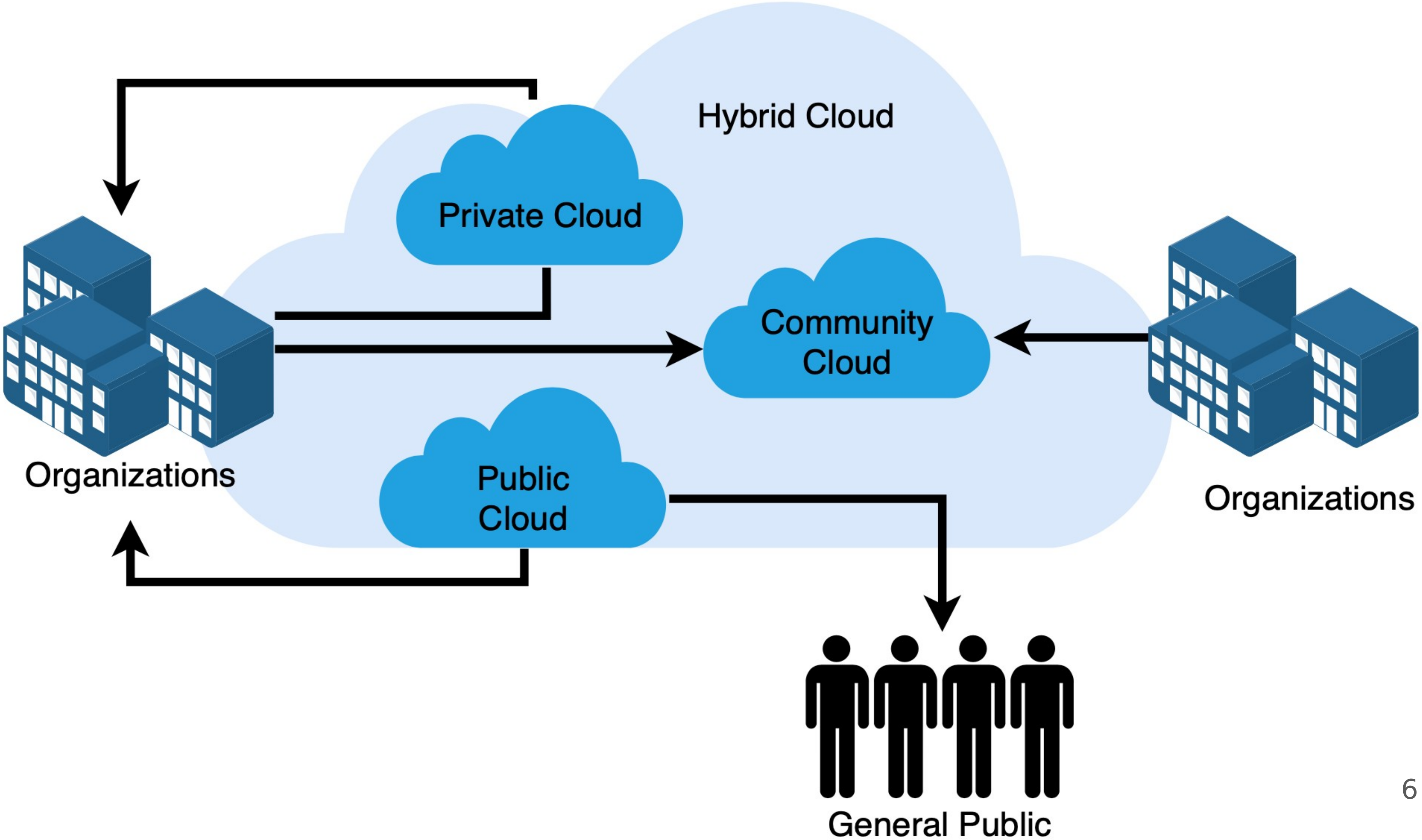
Introduction

Dependability demands for edge and fog computing are not precisely specified in the literature, as edge and fog computing are recent paradigms. The existing literature shows that authors still diverge considerably from each other in the types of failures that they discuss, the technique employed, and even the dependability demands themselves.

Background

Centro de Informática UFPE | Universidade Federal de Pernambuco

Cloud Computing at a Glance



Fog Computing at a Glance

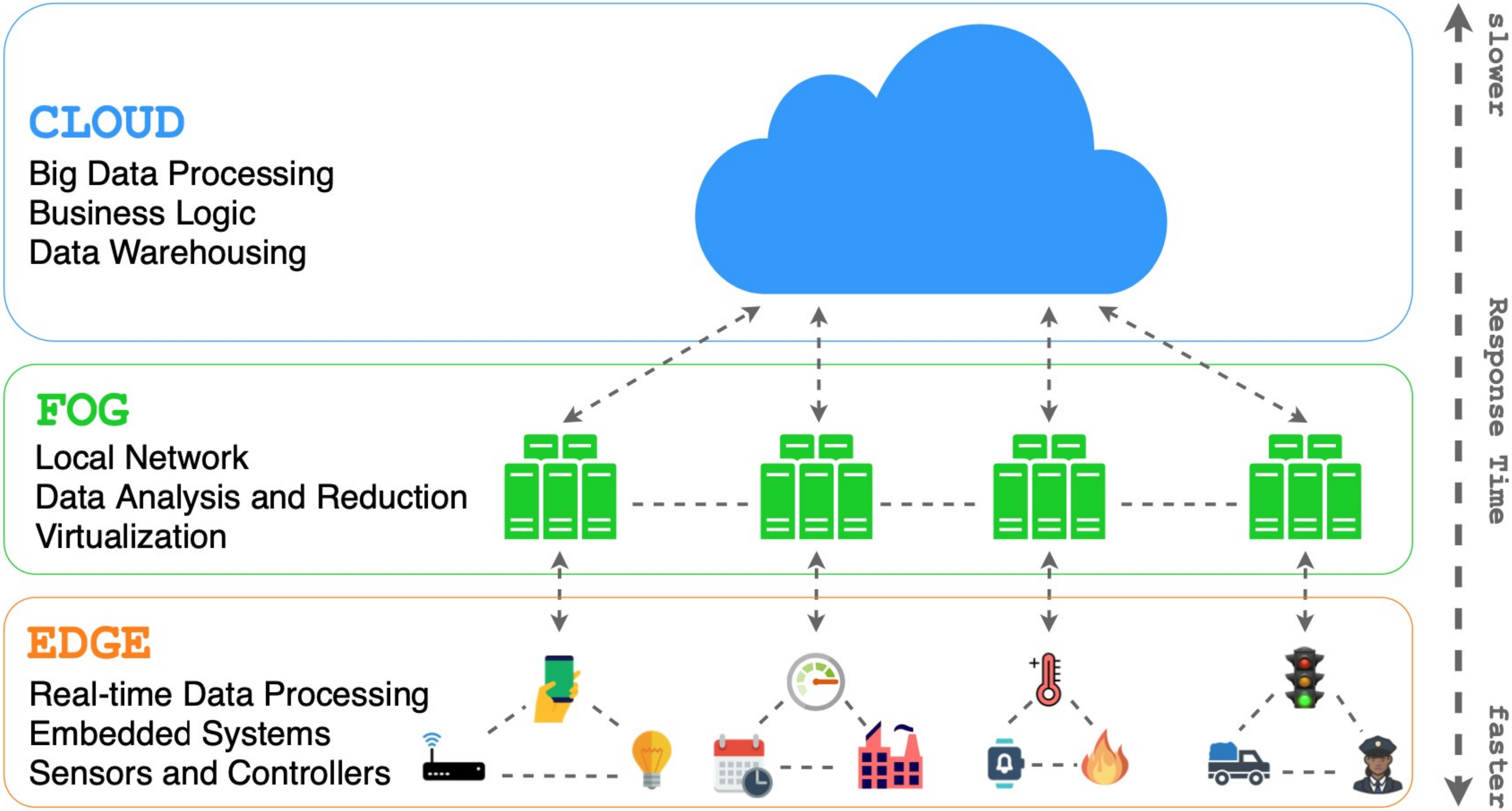
Fog computing is a term proposed by researchers from Cisco System, and it is related to the processing power at the edge of the network. There is also a similar concept called cloudlets; however, cloudlets are related to the mobile network; but fog computing is commonly related to IoT infrastructure.

Fog Computing at a Glance

- **reduction of network traffic** - fog environments drastically reduces the traffic being sent to the cloud because most of the data is processed closer to the end devices.
- **suitable for IoT tasks and queries** - by using fog environments, IoT tasks may be processed closer to the physical location of the sensors/actuators, which makes the application closer to the geographical context of the sensor or actuator.
- **low latency requirement** - the data processing is performed closer to the nodes; thus, it makes real-time response possible.

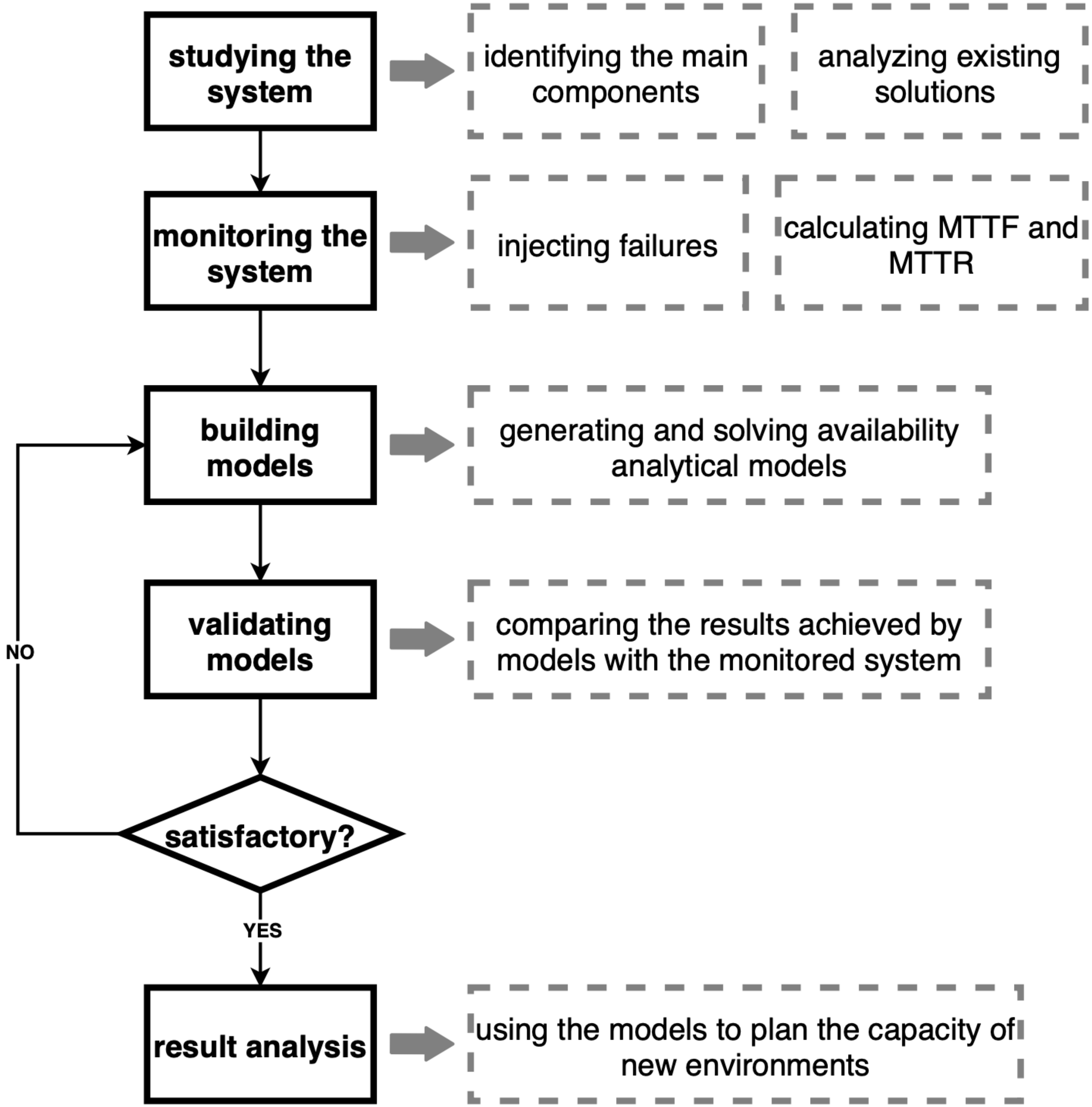
Edge Computing at a Glance

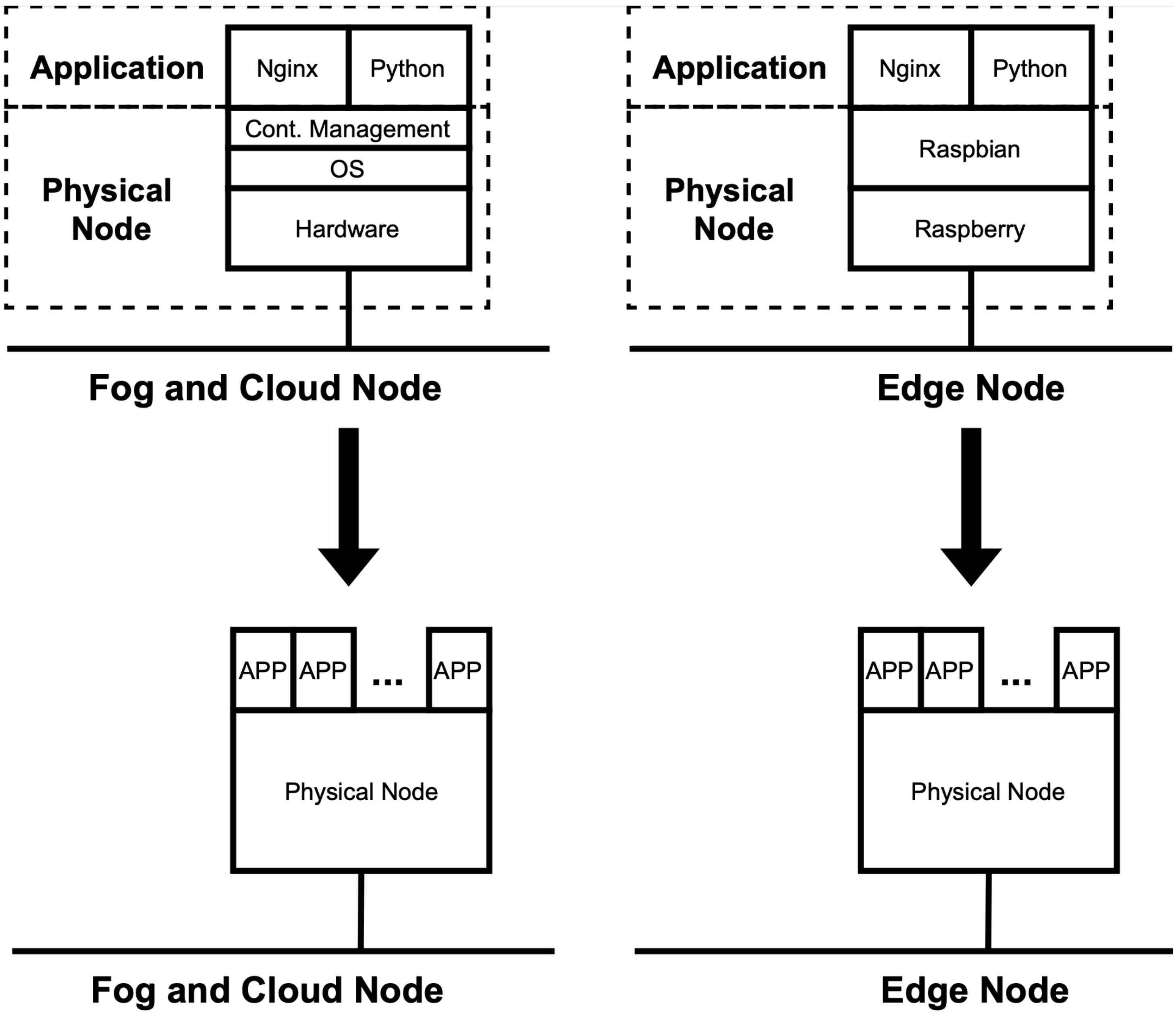
Edge computing paradigm processes the data locally at the edge nodes; then the selected data or insights flow to the cloud server. This is the opposite of the traditional cloud paradigm, which allows all the raw data generated from the edge devices to flow to the cloud server.


CLOUD
Big Data Processing
Business Logic
Data Warehousing
FOG
Local Network
Data Analysis and Reduction
Virtualization
EDGE
Real-time Data Processing
Embedded Systems
Sensors and Controllers
slower
Response Time
faster

An Overview of the Proposal

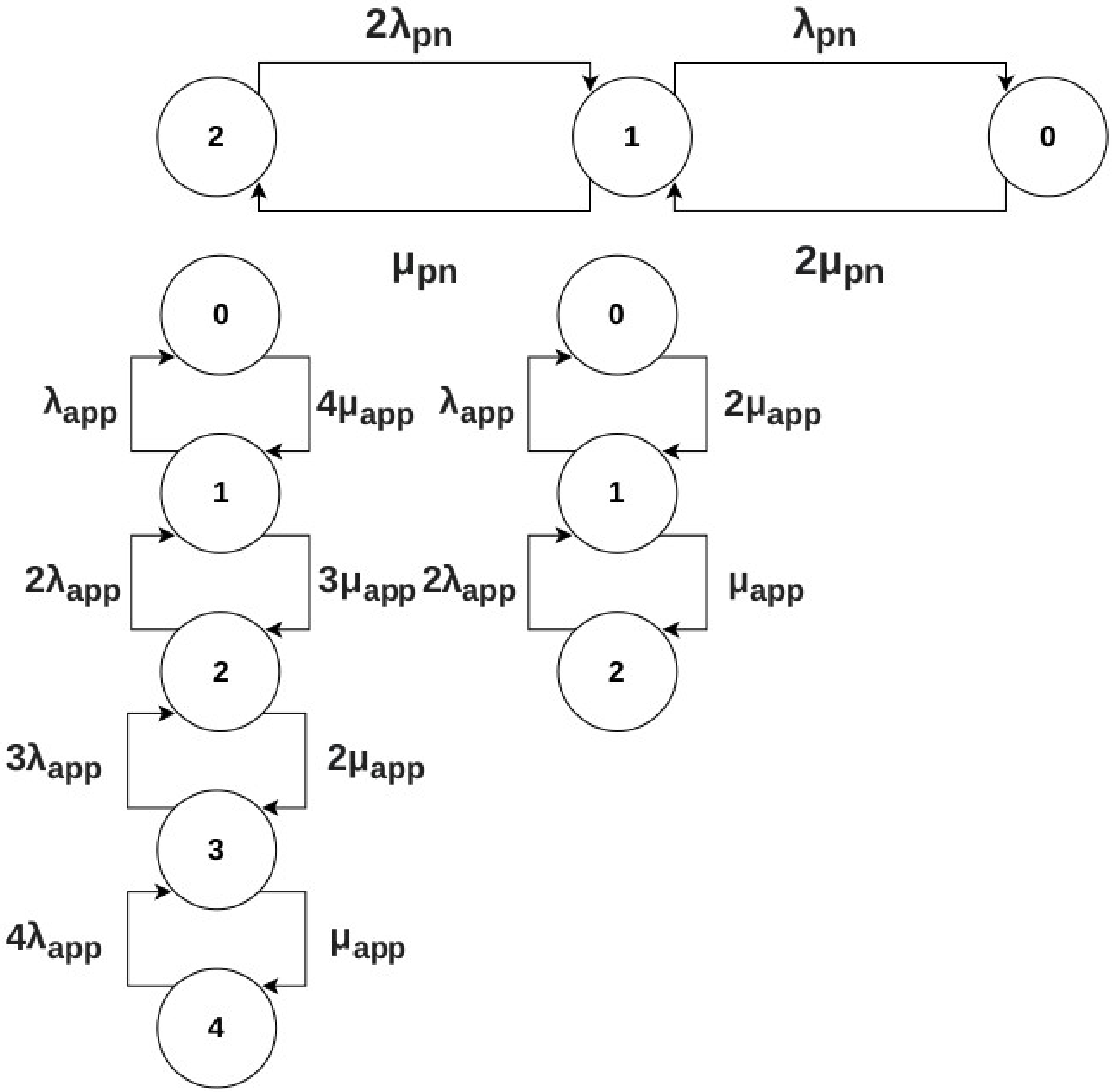
Centro de Informática UFPE
Universidade Federal de Pernambuco

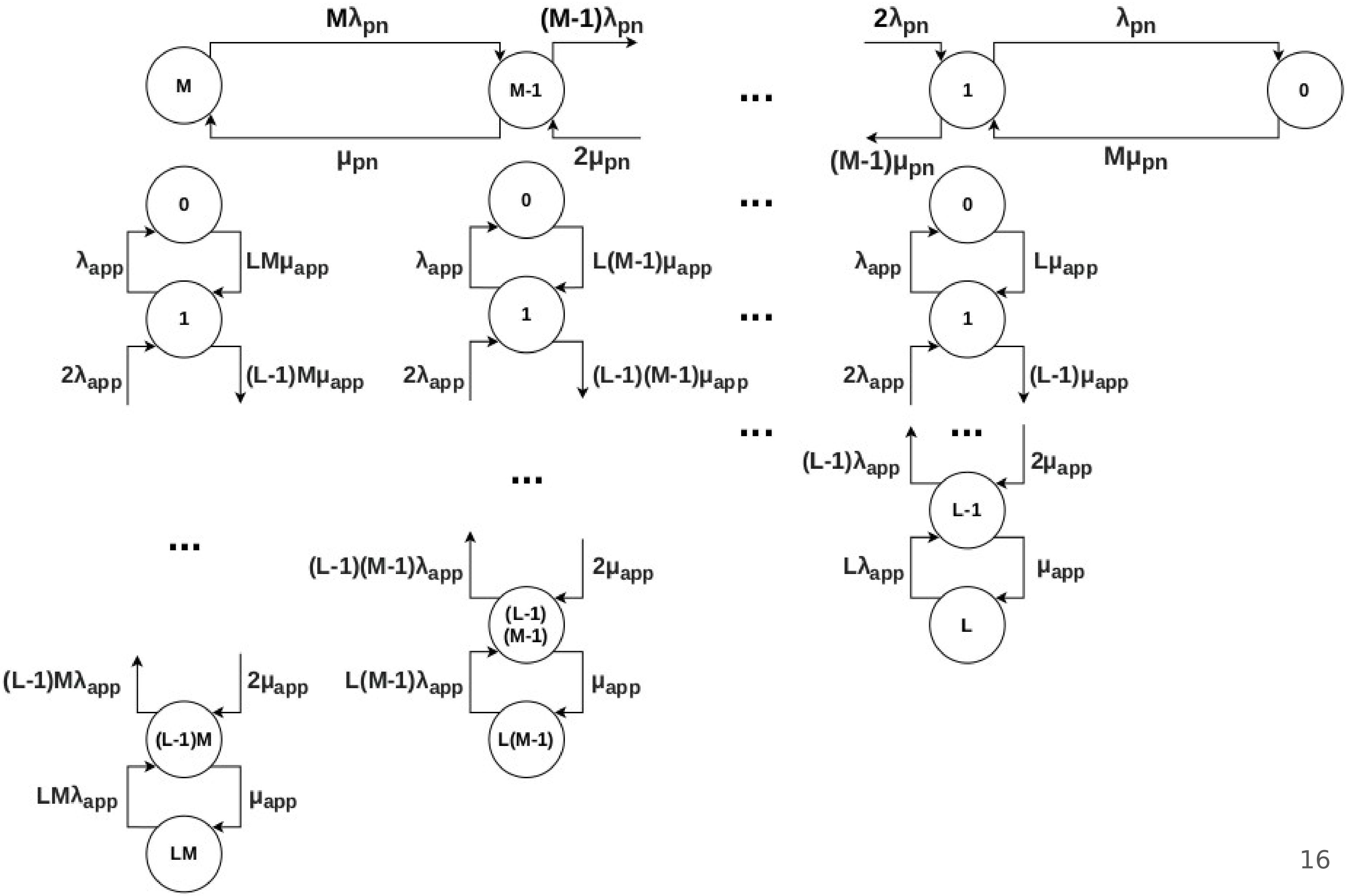

studying the system
identifying the main components
analyzing existing solutions
monitoring the system
injecting failures
calculating MTTF and MTTR
building models
generating and solving availability analytical models
validating models
comparing the results achieved by models with the monitored system
NO
satisfactory?
YES
result analysis
using the models to plan the capacity of new environments


Application
Nginx
Python
Physical Node
Cont. Management
OS
Hardware
Fog and Cloud Node
Application
Nginx
Python
Physical Node
Raspbian
Raspberry
Edge Node
APP
APP
...
APP
Physical Node
Fog and Cloud Node
APP
APP
...
APP
Physical Node
Edge Node

Availability Models

Through the use of analytical availability models, we are able to overcome the two principal constraints of measurement-based evaluation: it is less expensive, and it may be used during the design phase of the system (GOKHALE; TRIVEDI , 1998). Using analytical models, we can predict and plan how the system will behave before deploying it. Therefore, we can develop better systems.


2λpn
λpn
2
1
0
μpn
2μpn
0
λapp
4μapp
1
2λapp
3μapp
2
3λapp
2μapp
3
4λapp
μapp
4
0
λapp
2μapp
1
2λapp
μapp
2


$M\lambda_{pn}$
$(M-1)\lambda_{pn}$
$2\lambda_{pn}$
λ_{pn}
M
M-1
1
0
...
μ_{pn}
$2\mu_{pn}$
$(M-1)\mu_{pn}$
$M\mu_{pn}$
0
1
λ_{app}
$LM\mu_{app}$
$2\lambda_{app}$
$(L-1)M\mu_{app}$
$L(M-1)\mu_{app}$
$(L-1)(M-1)\mu_{app}$
$L\mu_{app}$
$(L-1)\mu_{app}$
$(L-1)\lambda_{app}$
$2\mu_{app}$
L-1
$L\lambda_{app}$
μ_{app}
L
$(L-1)(M-1)\lambda_{app}$
(L-1)
(M-1)
$L(M-1)\lambda_{app}$
L(M-1)
$(L-1)M\lambda_{app}$
(L-1)M
$LM\lambda_{app}$
LM

Availability Model

$$A = \sum_{i=1}^{M} \left(\left(1 - \frac{\lambda_{app}^{iL}}{(\lambda_{app} + \mu_{app})^{iL}} \right) \times \frac{M!}{i!\,(M-i)!} \times \frac{\lambda_{pn}^{M-i}\,\mu_{pn}^{i}}{(\lambda_{pn} + \mu_{pn})^{M}} \right).$$

Availability K out of N Model

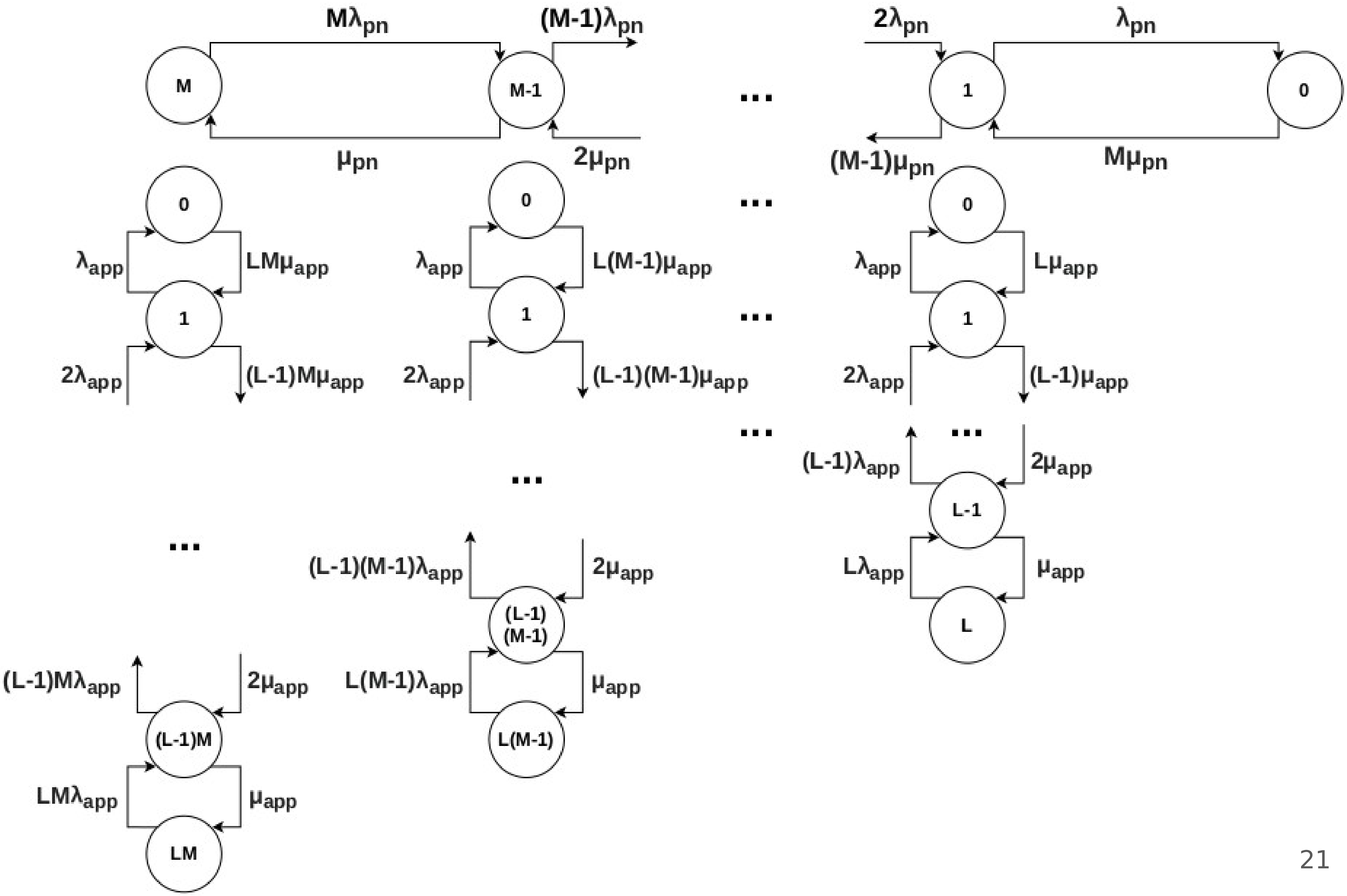
$$A_{KooN} = \sum_{i=1}^{M} \left(\left(\sum_{j=K}^{iL} \frac{(iL)!}{j!\ (iL-j)!} \times \frac{\lambda_{app}^{iL-j}\ \mu_{app}^{j}}{(\lambda_{app} + \mu_{app})^{iL}} \right) \times \left(\frac{M!}{i!\ (M-i)!} \times \frac{\lambda_{pn}^{M-i}\ \mu_{pn}^{i}}{(\lambda_{pn} + \mu_{pn})^{M}} \right) \right).$$

Average number of applications available

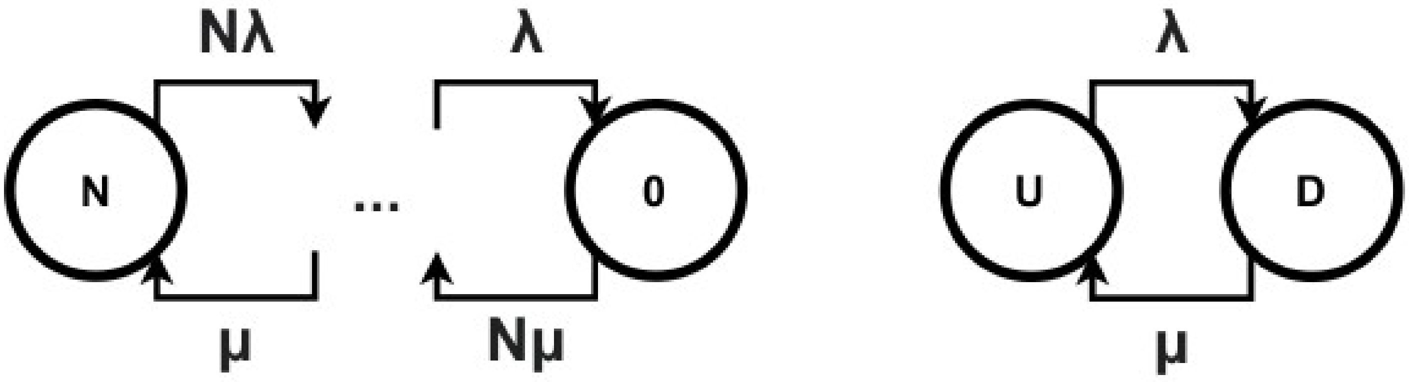
$$ANAPPA = \sum_{i=1}^{M} \sum_{j=1}^{i \times L} \left(\frac{j (iL)!\ \mu_{app}^{j}\ \lambda_{app}^{iL-j}\ (\lambda_{app} + \mu_{app})^{-iL}}{j!\ (iL - j)!} \times \frac{M!\ \mu_{pn}^{i}\ \lambda_{pn}^{M-i}\ (\lambda_{pn} + \mu_{pn})^{-M}}{i!\ (M - i)!} \right).$$

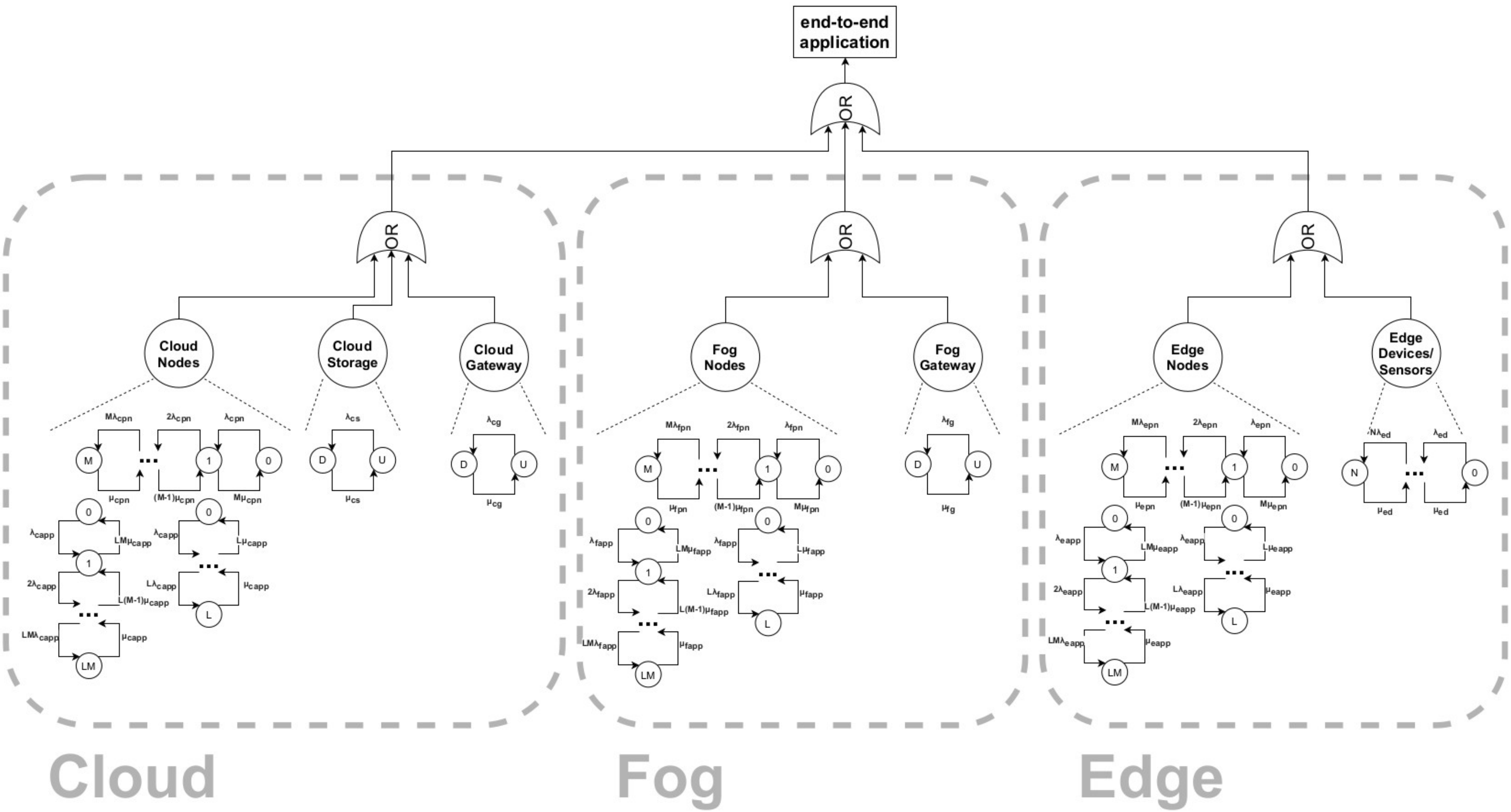
Hierarchical Availability Model

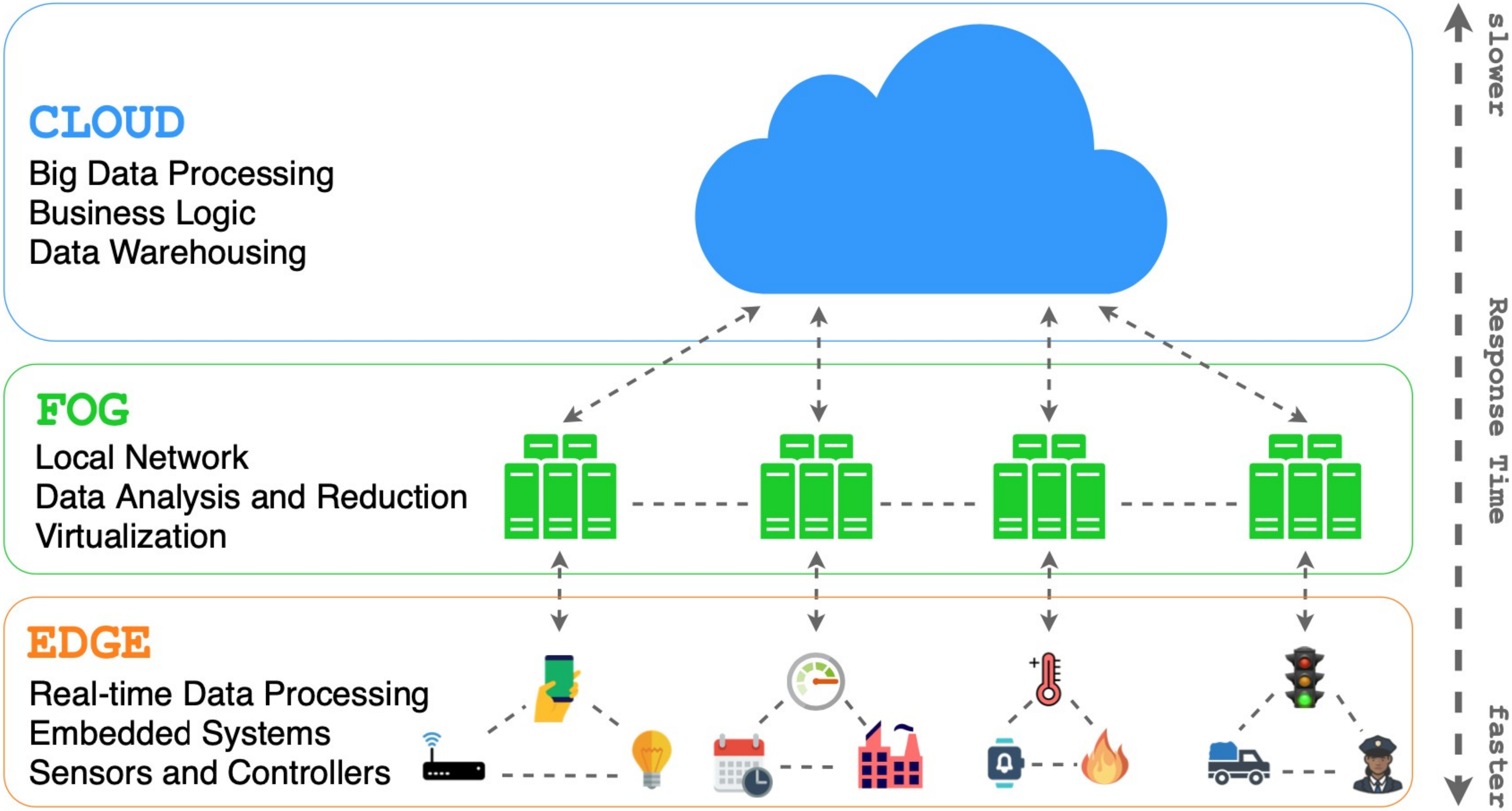
Using hierarchical models, we might represent and design more accurately an environment, and then we might use it to predict the behavior of the infrastructure of our service. In this section, we present the hierarchical model representing the availability metric of edge-fog-cloud environment. To start, we depict a Markov chain, which is a generalization of the components in the infrastructure. Later, we use the Markov chain to obtain relevant closed-form solutions to calculate the availability metrics of the components.


Mλpn
(M-1)λpn
2λpn
λpn
M
M-1
...
1
0
μpn
2μpn
(M-1)μpn
Mμpn
0
0
0
λapp
LMμapp
λapp
L(M-1)μapp
λapp
Lμapp
1
1
1
2λapp
(L-1)Mμapp
2λapp
(L-1)(M-1)μapp
2λapp
(L-1)μapp
...
(L-1)λapp
2μapp
L-1
Lλapp
μapp
L
(L-1)(M-1)λapp
2μapp
(L-1)
(M-1)
L(M-1)λapp
μapp
L(M-1)
(L-1)Mλapp
2μapp
(L-1)M
LMλapp
μapp
LM

Hierarchical Availability Model



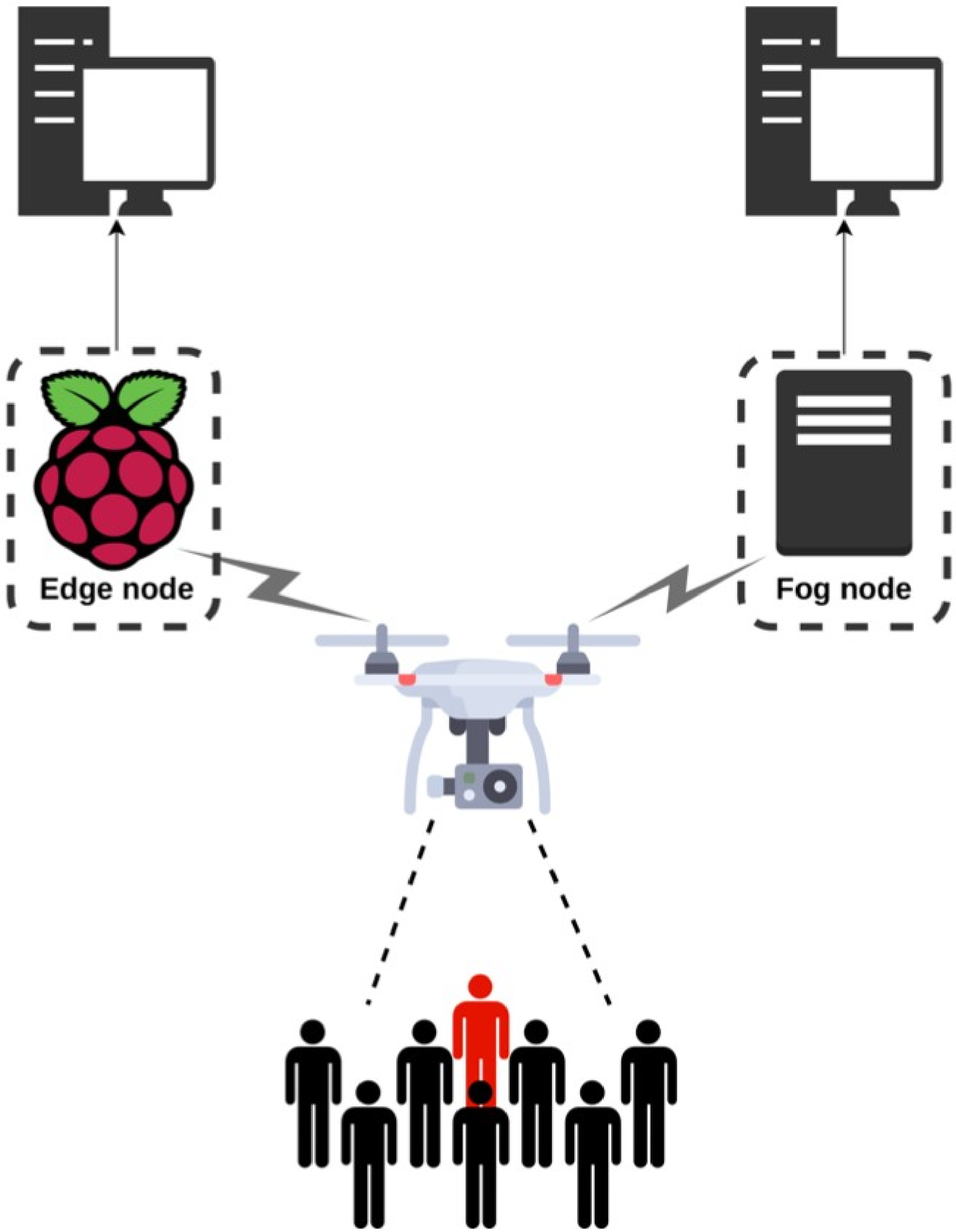

end-to-end application
OR
OR
OR
OR
Cloud Nodes
Cloud Storage
Cloud Gateway
Fog Nodes
Fog Gateway
Edge Nodes
Edge Devices/ Sensors
Cloud
Fog
Edge

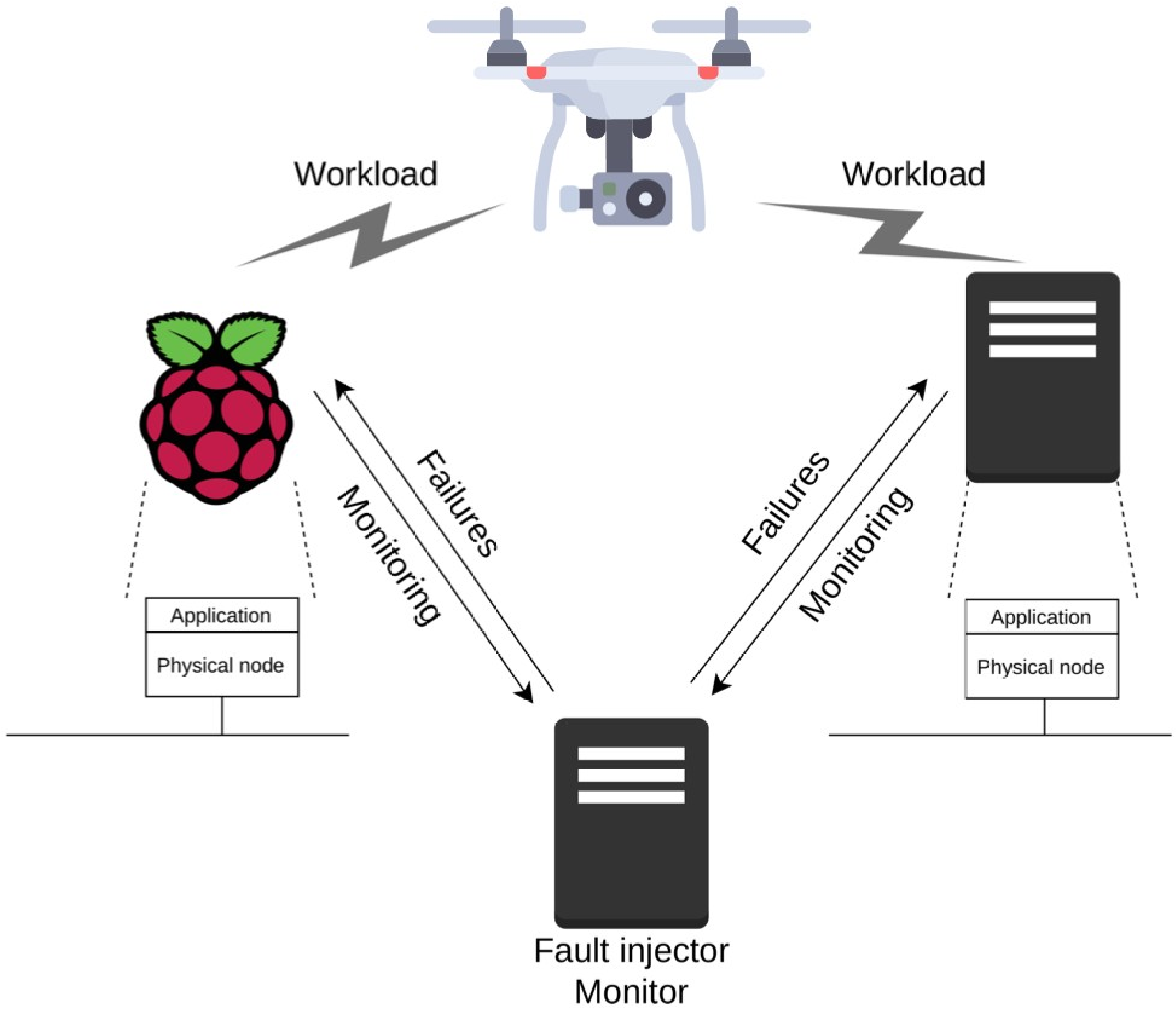

CLOUD
Big Data Processing
Business Logic
Data Warehousing
FOG
Local Network
Data Analysis and Reduction
Virtualization
EDGE
Real-time Data Processing
Embedded Systems
Sensors and Controllers
slower
Response Time
faster

Case Studies

Case Study I - Availability Model Validation

In this case study, we describe the experiments to evaluate whether our availability models represent the real environment or not; our experimental environment was composed of one drone, three machines (fog node and two clients), and one Raspberry (edge node). The drone is a Dji Mavic Pro Platinum and is responsible for capturing video and sending it to the fog and edge infrastructure.


Edge node
Fog node


Workload
Workload
Failures
Monitoring
Failures
Monitoring
Application
Physical node
Application
Physical node
Fault injector
Monitor

Case Study I - Availability Model Validation

Node	Components	MTTF (h)	MTTR (h)
Edge	Raspberry	4767.8	3.48
	OS	2880	1
	Python App	217.8	0.46
	Nginx App	217.8	0.46
Fog	Hardware	8760	1.67
	OS	2880	1
	Cont. Management	2880	1
	Python App	217.8	0.46
	Nginx App	217.8	0.46

Node	Components	MTTF (h)	MTTR (h)
Edge	Raspberry	5.4	3.48
	OS	3.3	1
	Python App	0.9	0.46
	Nginx App	0.9	0.46
Fog	Hardware	10	1.67
	OS	3.3	1
	Cont. Management	3.3	1
	Python App	0.9	0.46
	Nginx App	0.9	0.46

Case Study I - Availability Model Validation

Node	Components	MTTF (h)	MTTR (h)
Edge	Physical Node	2.04	2.34
	Application	0.34	0.44
Fog	Physical Node	2.48	1.29
	Application	0.34	0.44

Case Study I - Availability Model Validation

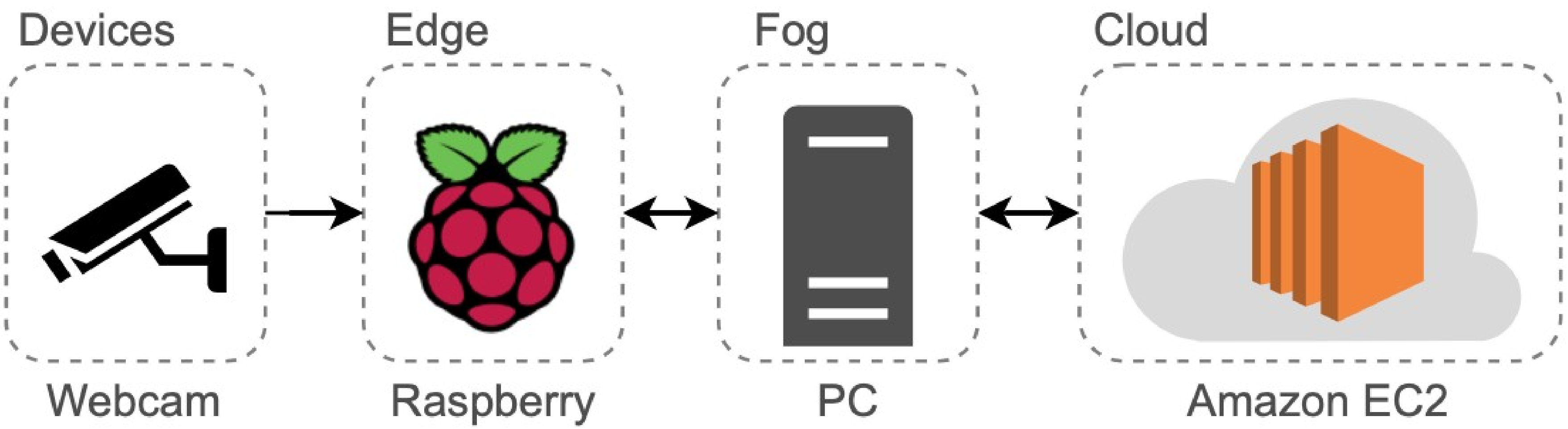
Table 8 – Validation result.

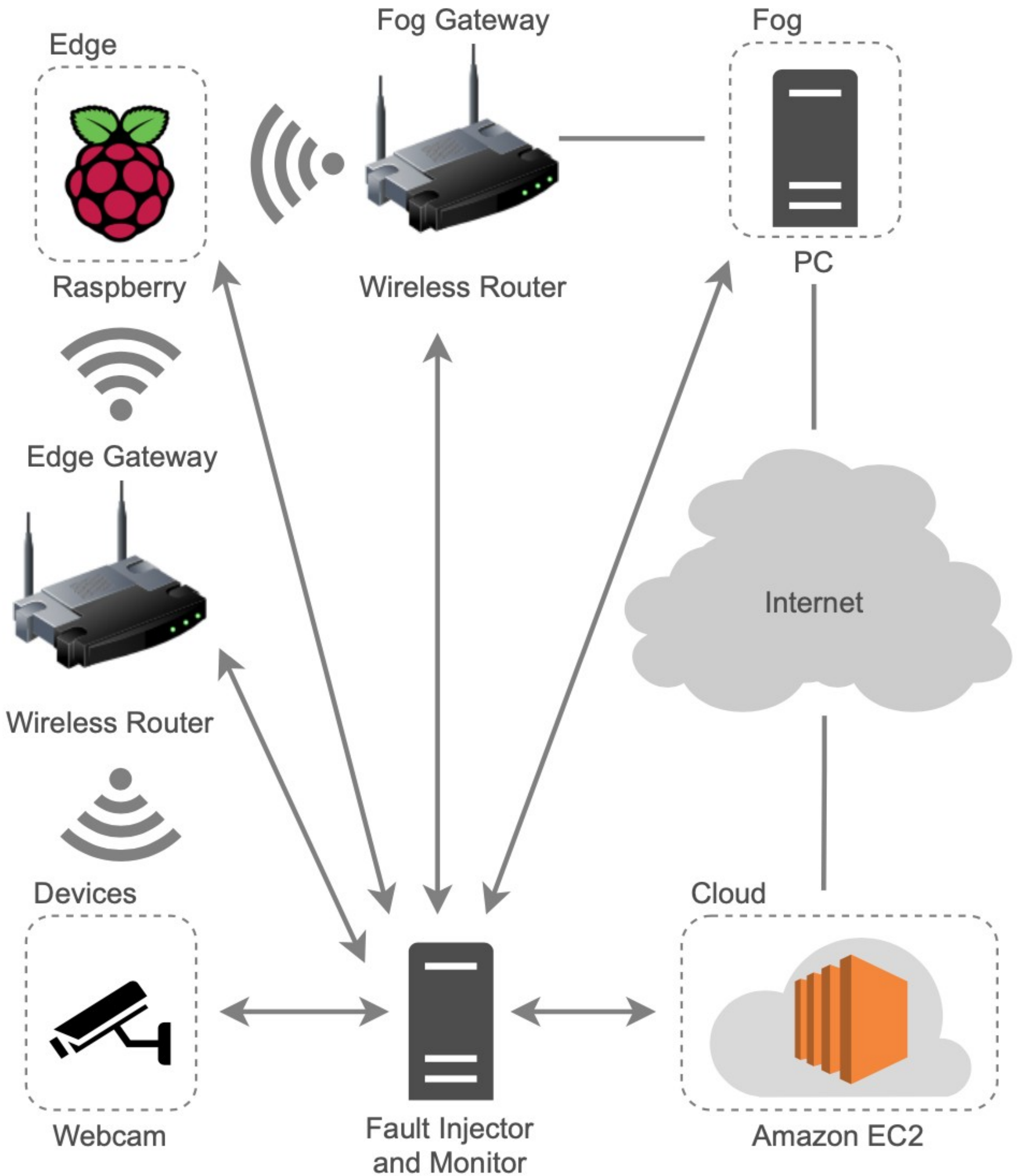
Node	Model Result	System Measurement	CI 95% of the system
Edge	0.2030	0.2215	$0.1972 < \theta < 0.2459$
Fog	0.2867	0.2598	$0.2230 < \theta < 0.2966$

Case Study II - Hierarchical Model Validation

In this case study, we describe the experiments to evaluate whether our hierarchical availability model represents the real environment or not; our experimental environment comprised one webcam, a Raspberry Pi (edge node), and one personal computer (fog node). The webcam is responsible for capturing video and sending it to the edge node, which is responsible for doing the first pre-processing activity; the edge sends it to the fog, and then the fog send it to the cloud infrastructure. The video captured by the webcam, which is an IP Camera D-LINK H.264 DCS-931L, with 30 frames per second and a resolution of 720×480.

Case Study II - Hierarchical Model Validation




Edge
Raspberry
Fog Gateway
Wireless Router
Fog
PC
Edge Gateway
Wireless Router
Internet
Devices
Webcam
Fault Injector
and Monitor
Cloud
Amazon EC2

Case Study II - Hierarchical Model Validation

Model Result	System Measurement	CI 95% of the system
0.2434	0.2312	$0.1838 < \theta < 0.2589$

prps@cin.ufpe.br

Conclusion

Centro de Informática
UFPE
Universidade Federal de Pernambuco

Conclusion

In these papers, we proposed hierarchical and analytical models to evaluate the availability of the edge-fog-cloud continuum. We performed two case studies to show how our models can evaluate edge-fog-cloud systems, where we built an intelligent traffic management baseline infrastructure. This infrastructure is composed of a camera, edge node, fog node, and a cloud environment.

prps@cin.ufpe.br

Thanks for listening!

Centro de Informática UFPE

Universidade Federal de Pernambuco

References

- BATTULA, S. K.; O'REILLY, M. M.; GARG, S.; MONTGOMERY, J. A generic stochastic model for resource availability in fog computing environments. IEEE Transactions on Parallel and Distributed Systems, IEEE, v. 32, n. 4, p. 960–974, 2020.
- BONOMI, F.; MILITO, R.; ZHU, J.; ADDEPALLI, S. Fog computing and its role in the internet of things. In: Proceedings of the first edition of the MCC workshop on Mobile cloud computing. [S.l.: s.n.], 2012. p. 13–16.
- GOKHALE, S. S.; TRIVEDI, K. S. Analytical modeling. Encyclopedia of Distributed Systems, Citeseer, 1998.
- HUO, W.; LIU, F.; WANG, L.; JIN, Y.; WANG, L. Research on distributed power distribution fault detection based on edge computing. IEEE Access, IEEE, v. 8, p. 24643–24652, 2019.
- KAFHALI, S. E.; SALAH, K. Efficient and dynamic scaling of fog nodes for iot devices. The Journal of Supercomputing, Springer, v. 73, n. 12, p. 5261–5284, 2017.

References

- MACIEL, P.; TRIVEDI, K.; MATIAS, R.; KIM, D. Dependability modeling. In: Performance and Dependability in Service Computing: Concepts, Techniques and Research Directions. [S.l.: s.n.], 2011.
- MATOS, R.; ARAUJO, J.; OLIVEIRA, D.; MACIEL, P.; TRIVEDI, K. Sensitivity analysis of a hierarchical model of mobile cloud computing. Simulation Modelling Practice and Theory, Elsevier, v. 50, p. 151–164, 2015.
- MATOS, R.; DANTAS, J.; ARAUJO, J.; TRIVEDI, K. S.; MACIEL, P. Redundant eucalyptus private clouds: Availability modeling and sensitivity analysis. J. Grid Comput., Springer-Verlag, v. 15, n. 1, p. 1–22, mar. 2017. ISSN 1570-7873.
- PATIL, V.; ASHRA, J.; GAJINKAR, M.; FAIZEE, S. A review: Office monitoring and surveillance system. Available at SSRN 3561737, 2020.
- SILVA, B.; MATOS, R.; CALLOU, G.; FIGUEIREDO, J.; OLIVEIRA, D.; FERREIRA, J.; DANTAS, J.; LOBO, A.; ALVES, V.; MACIEL, P. Mercury: An integrated environment for performance and dependability evaluation of general systems. In: Proceedings of Industrial Track at 45th Dependable Systems and Networks Conference, DSN. [S.l.: s.n.], 2015.