E2C*lab*: Reproducible Analysis of Complex Workflows on the Edge-to-Cloud Continuum

Daniel Rosendo

University of Rennes Inria, CNRS, IRISA - Rennes, France daniel.rosendo@inria.fr

CLOUD

centralized

Advisors:

Gabriel Antoniu

University of Rennes Inria, CNRS, IRISA Rennes, France

Alexandru Costan

University of Rennes Inria, CNRS, IRISA Rennes, France

Patrick Valduriez

University of Montpellier Inria, CNRS, LIRMM Montpellier, France

Introduction

Context

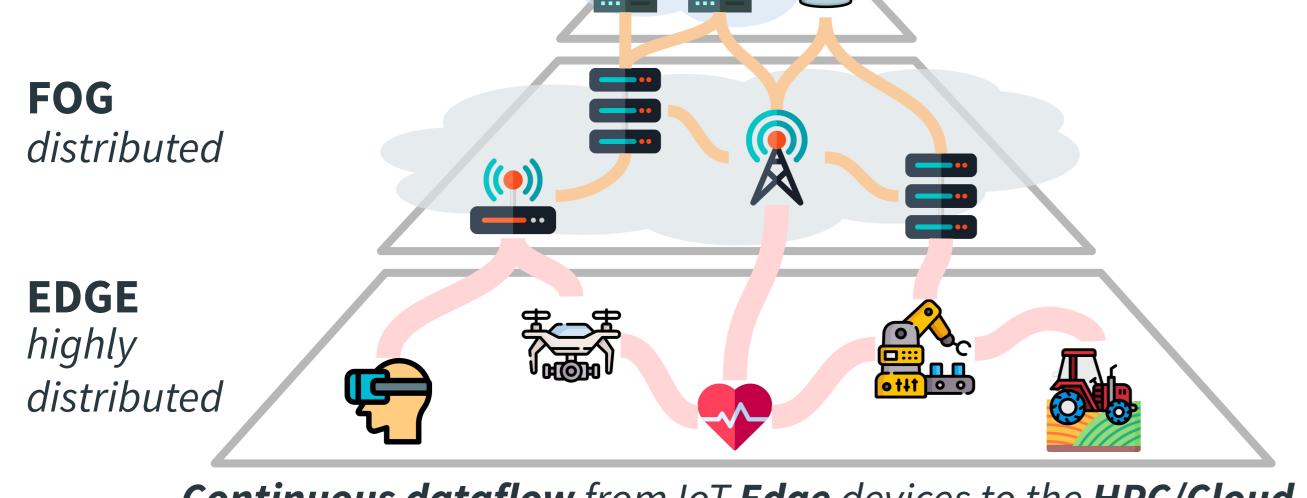
Complex Application Workflows on the Edge-to-Cloud Computing Continuum

Problem

How to systematically perform large-scale **experiments** to enable the **reproducibility** of the **results?**

Challenges

Establish a rigorous **methodology** for **reproducibility** of experiments



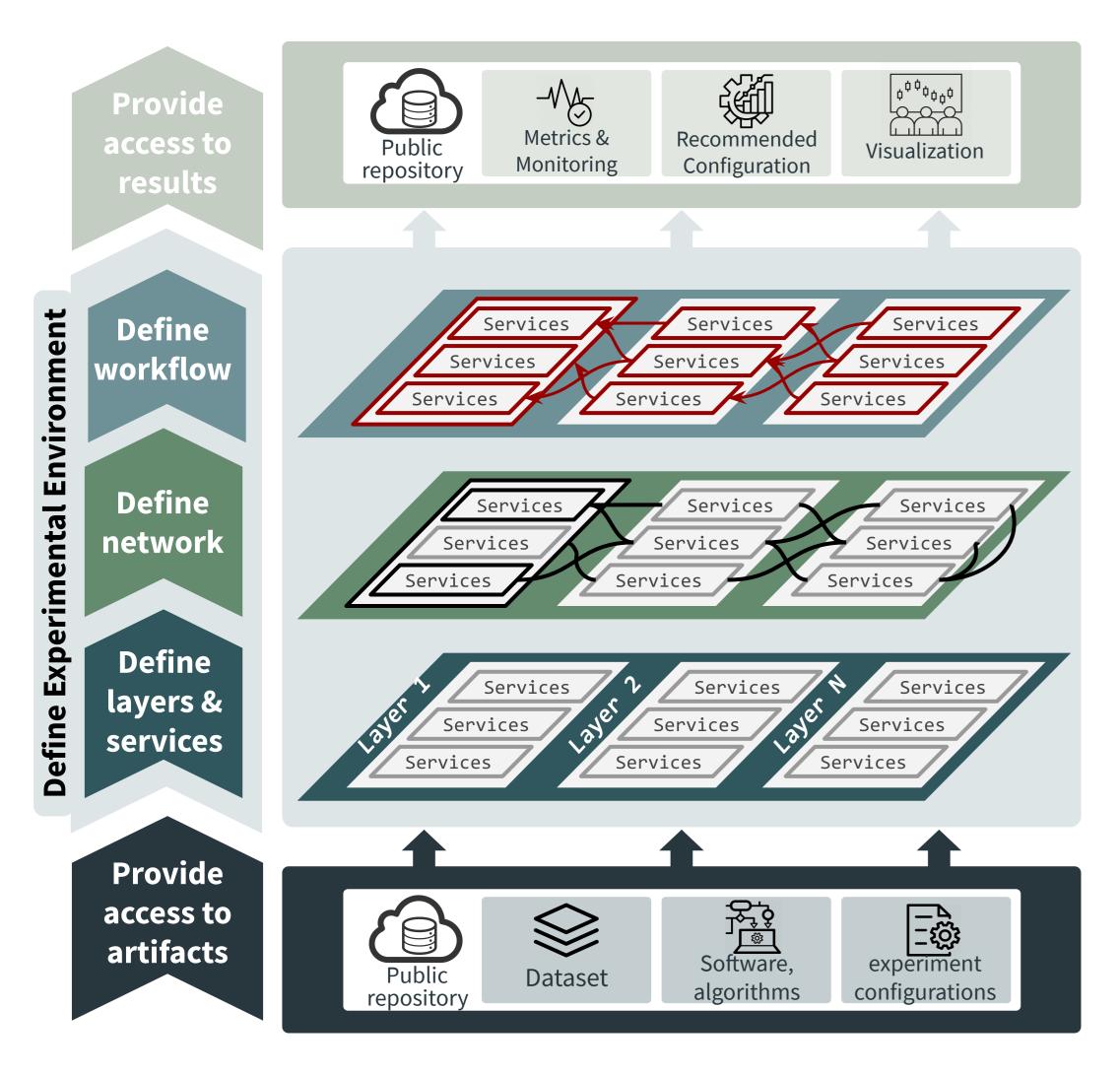
Continuous dataflow from IoT **Edge** devices to the **HPC/Cloud**

Which system parameters and infrastructure configurations impact performance and how?

Where should application parts be executed to minimize end-to-end latency? Achieve a **representative testbed setup** for a holistic **understanding of performance** of applications

Optimize application workflows on heterogeneous resources results in a **complex search space**

Our Contribution



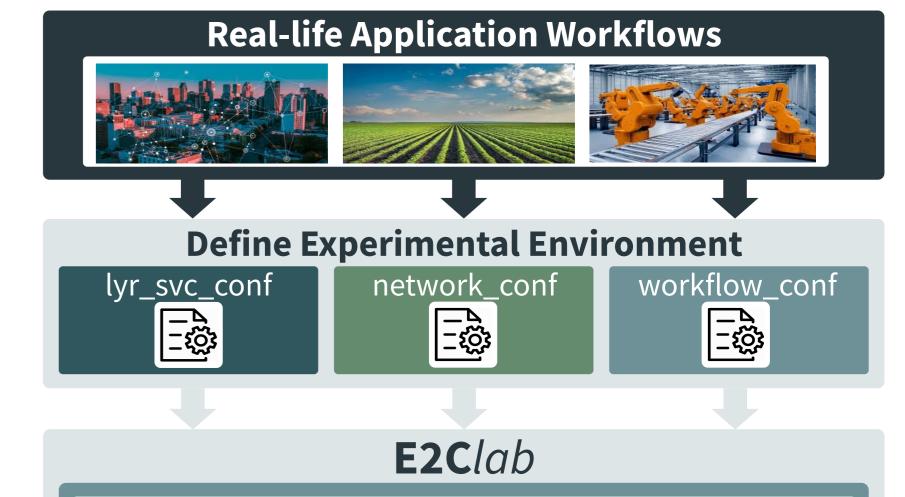


Reproducible Experiments

Repeatability, Replicability & Reproducibility

Mapping

Application parts & physical testbed







Variation & Scaling *Experiment variation and transparent scaling*

Network Emulation

Edge-to-Cloud

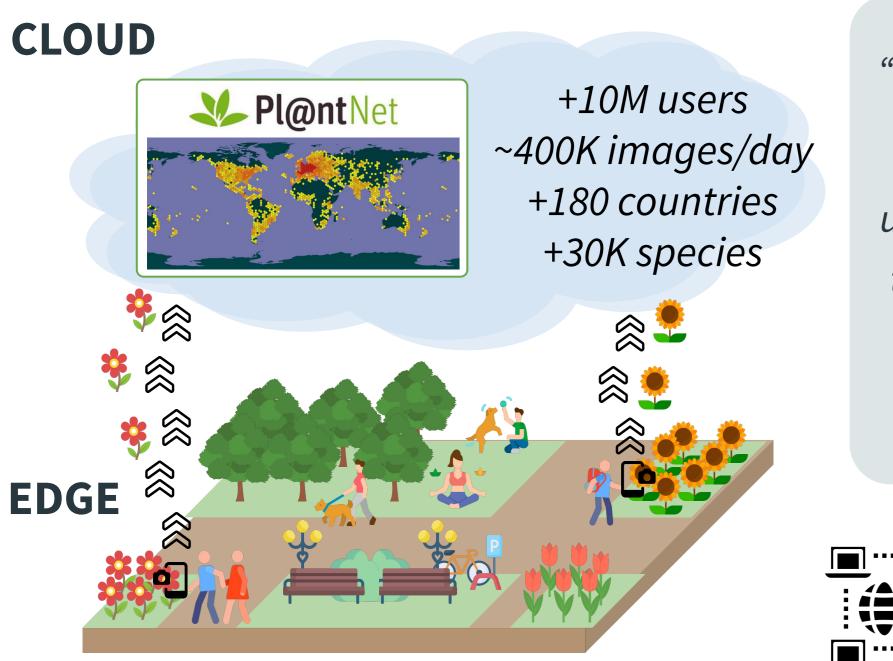
communication constraints

Application Optimization

Parameter search of application and infrastructure configurations

Experiment Manager					
$\begin{array}{c} \bullet \\ \bullet $					
LYR & SVC Manager	Network Manager	Workflow Manager	Optimization Manager		
÷					
EnOSlib					
Testbed Environments					

Validation with a Large-scale Real-life Application: Pl@ntNet



"Over **3-4 seconds** more than **60%** of users **abandon** the transaction and may even **delete** the application" What is the software configuration that minimizes the user response time?

Find	(http, download, simsearch, extract), in order to	
Minimize	UserResponseTime	
Subject to	20 ≤ (http, download, simsearch) ≤ 60, Pool Size.	
	$3 \le (extract) \le 9$, Pool Size.	

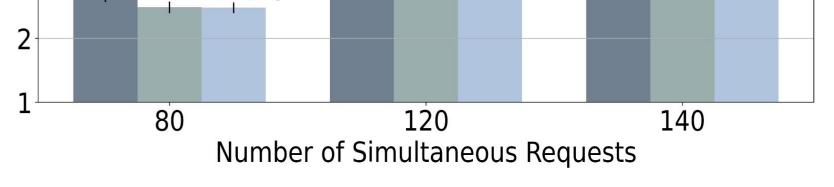
Thread pool	baseline	preliminary optimum	refined optimum	
HTTP	40	54	54	
Download	40	54	54	

How does the number of simultaneous users accessing the application impact on the user response time?

7 1	User Response Time					
6	Baseline Preliminary optimum	1	Refined op	otimum		
(sec)			4	.809		4,381
s) 4 amii 3	3.8	64	3.628			
<u> </u>	2.657 2.476					



es	User resp. time (sec)	2.65	2.48	2.4
	Simsearch	40	53	53
	Extract	7	7	6



Takeaways

Our methodology has proved useful for **understanding** and **improving** the **performance** of a real life application used at very large-scale.

The configuration found by **E2C***lab* can **process 35% more** requests and presents a **smaller user response time** compared to the baseline.

Perspectives

E2C*lab* enables the Computing Continuum vision by allowing **reproducible** experimental **research** on **large-scale testbeds**.

E2C*lab* supports the complete experimental cycle across the Computing Continuum: **deployment**, **analysis**, **optimization**.



https://gitlab.inria.fr/E2Clab

https://plantnet.org/

