Quantitative Properties: Specification, Verification and Synthesis

Srđan Krstić

advisor:
prof. Carlo Ghezzi

co-advisor:
Dr. Domenico Bianculli

Quantitative Properties: Specification, Verification and Synthesis

Srđan Krstić

advisor:
prof. Carlo Ghezzi

POLITECNICO
MILANO

co-advisor:
Dr. Domenico Bianculli

Quantitative properties

“Properties that can be objectively expressed using numbers (quantities) with a precisely defined unit of measure.”

– Wikipedia

Quantitative properties

Response Time

Throughput

Availability

Features of Quantitative Properties

Aggregate transformation

Timing relation between events

"The average response time of a service must not exceed 30 milliseconds, if invoked by a premium customer"

Numerical bound

Features of Quantitative Properties

Multiplicity of events

“At most 3 VM allocations are allowed within 2 minute time window.”

Time bounded sequence of events

More Generally…

Quantitative Properties

$$x \bowtie 5$$

…express numerical bound on a certain value

…consider sequence of events

$$\overbrace{e_1 \rightarrow e_2 \rightarrow \ldots \rightarrow e_n}^{T}$$

bounded by absolute time T

$$e_1 \rightarrow e_2 \rightarrow \ldots \rightarrow p$$

bounded by an event

$$e_1 \rightarrow e_2 \rightarrow e_3 \rightarrow \ldots$$

or unbounded

More Generally…

Quantitative Properties

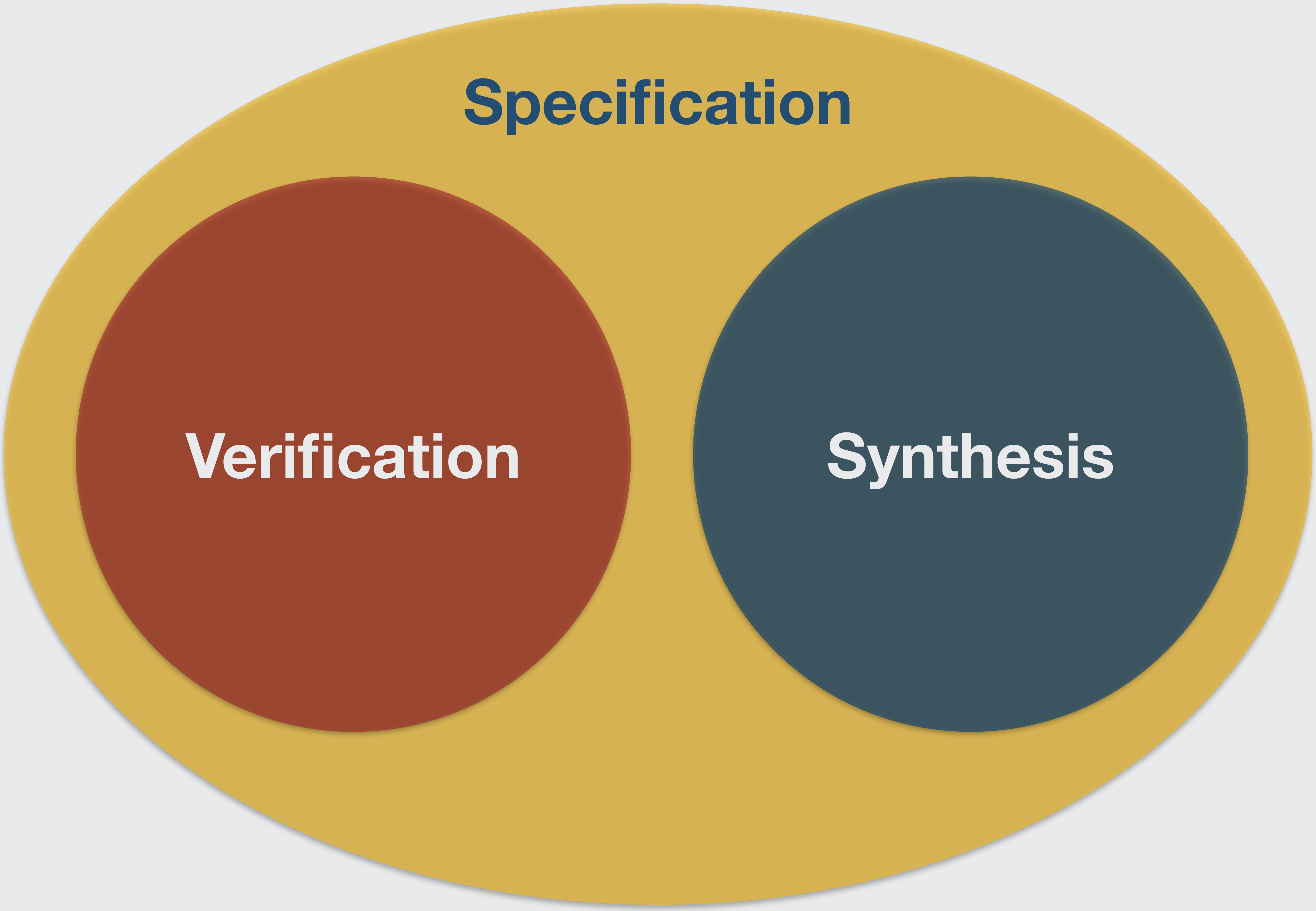
…compute numerical values from

$\{e_1, e_2, \ldots, e_n\}$ a set of specific events

$(e_1, e_2, \ldots, e_n)$ timing relations between tuples of specific events

$\{max(), avg(), count(), \ldots\}$ …apply aggregate transformations

Scope



Specification

1. Field Study

- Service-Based Applications [1]
- **Cloud-Based Systems**
- **Pervasive Systems**

- Research
- Practice

[1] Bianculli et al: Specification patterns from research to industry: a case study in service based applications. ICSE-SEIP 2012

2. Definition and Documentation

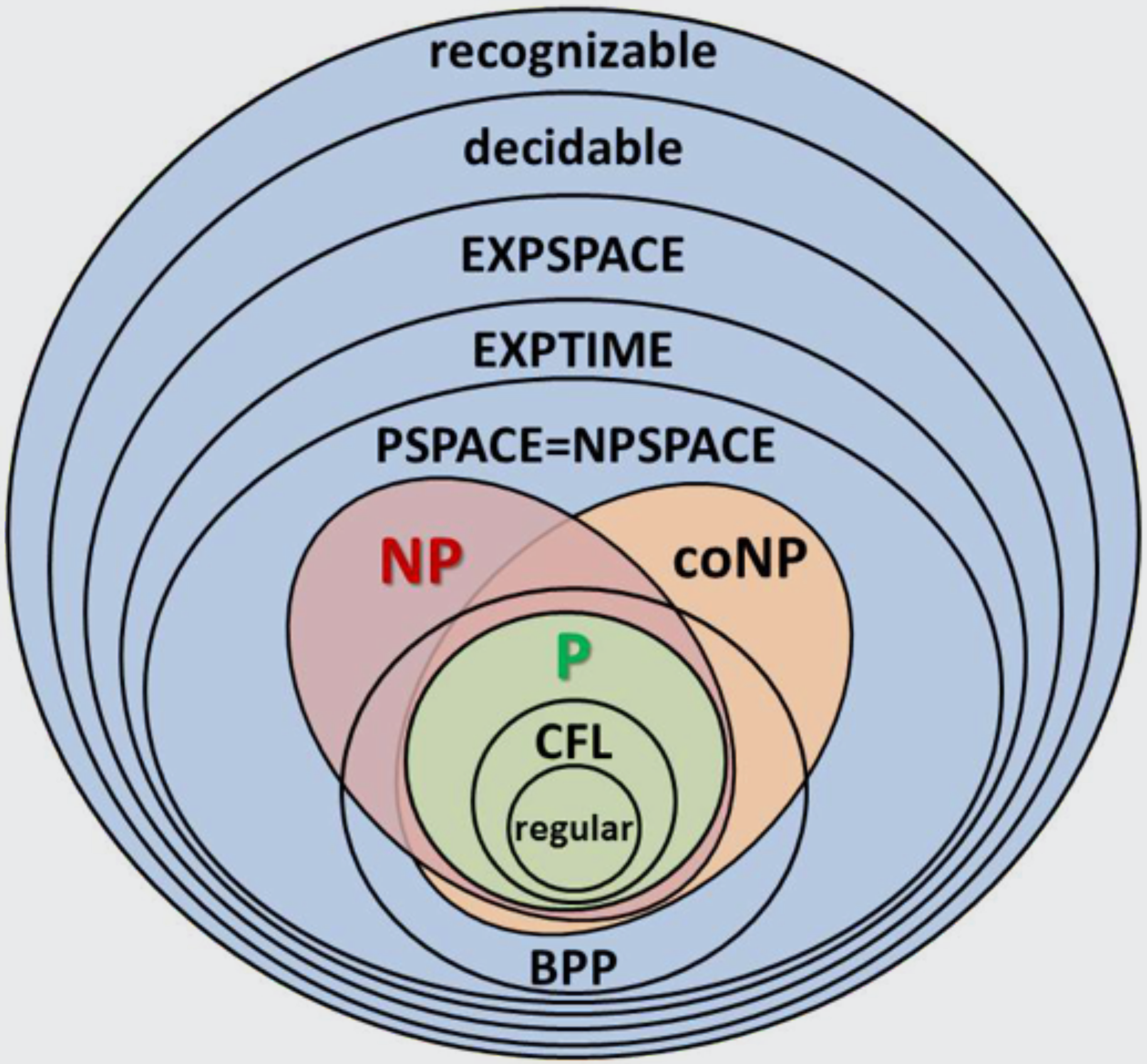
se·man·tics \si-ˈman-tiks\

noun plural but singular or plural in construction

- the study of the meanings of words and phrases in language
- the meanings of words and phrases in a particular context

3. Specification Patterns

4. Decidability and Complexity



5. Usability


CUSTOMER SERVICE
Please rate your experience
Excellent
Good
Average
Poor

Verification

1. Offline Trace Checking

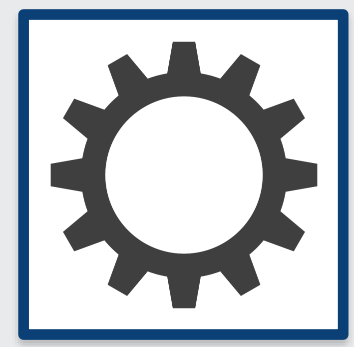
2. Runtime Verification

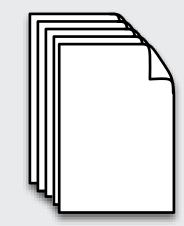
Synthesis

Synthesis

Logs

$$\left.\begin{array}{c}\phi_1^{K_1,\ldots,K_m}(p_1,\ldots,p_n)\\ \ldots\\ \phi_h^{K_1,\ldots,K_m}(p_1,\ldots,p_n)\end{array}\right\}$$





Specification templates

Inferred specifications: S1, S2, … , Sp

Progress

Specification

Elasticity

Resource Management

Quality of Service

Specification

Presented @ PESOS 2014

Towards the Formalization of Properties of Cloud-based Elastic Systems

Marcello M. Bersani
Politecnico di Milano
Milano, Italy
bersani@elet.polimi.it

Domenico Bianculli
University of Luxembourg
Luxembourg, Luxembourg
domenico.bianculli@uni.lu

Schahram Dustdar
TU Wien
Vienna, Austria
dustdar@infosys.tuwien.ac.at

Alessio Gambi
University of Lugano
Lugano, Switzerland
alessio.gambi@usi.ch

Carlo Ghezzi
Politecnico di Milano
Milano, Italy
carlo.ghezzi@polimi.it

Srđan Krstić
Politecnico di Milano
Milano, Italy
srdan.krstic@polimi.it

ABSTRACT

Cloud-based elastic systems run on a cloud infrastructure and have the capability of dynamically adjusting the alloca- ... of their resources in response to changes in the work- ... that balances the trade-off between the desired ... operational costs. The actual elas- ... mined by a combination ... of the elas-

1. INTRODUCTION

Cloud computing has become a practical solution to manage and leverage IT resources and services. Cloud platforms offer several benefits, among which the ability to access resources or service applications offered as (remote) services, available on-demand and on-the-fly, and billed according to a pay-per-use model.

Cloud providers offer resources and services at three different layers: at the *Software-as-a-Service* (*SaaS*) layer, users ... remotely access full-fledged software applications; at the ... *Service* (*PaaS*) layer, one finds a development ... and a run-time execution environ- ... provided code in sandboxes ... *structure-as-a-*

Verification

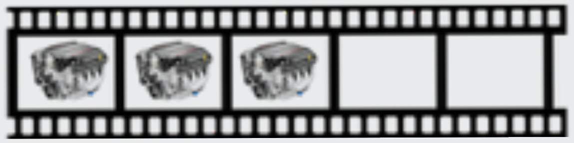
Specificati**O**n **L**anguage f**O**r Serv**I**ce Compo**S**ition In**T**eractions

Metric temporal logic with Aggregates [2]

[2] Bianculli et al.: The tale of SOLOIST: a specification language for service compositions interactions. FACS 2012

Verification

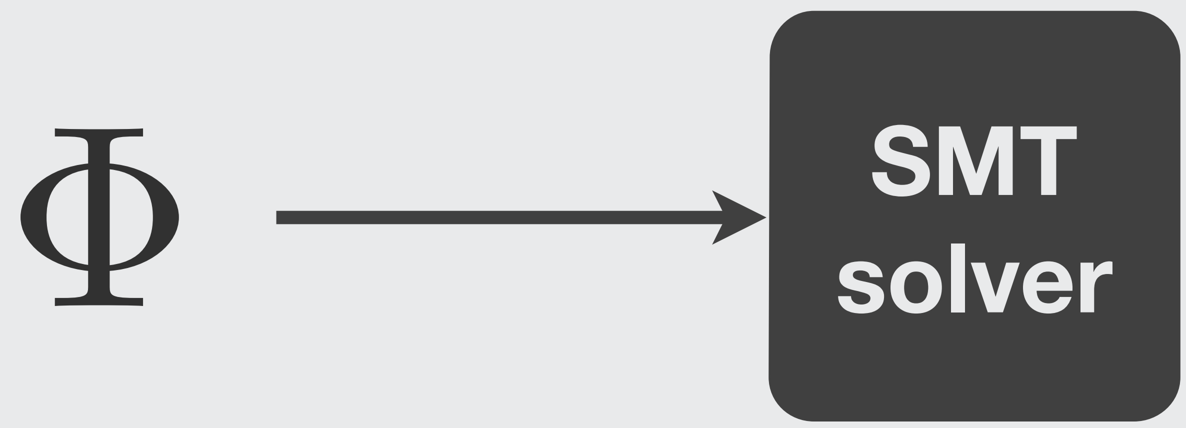
Execution trace **SOLOIST**

 $\wedge \quad \neg \varphi$

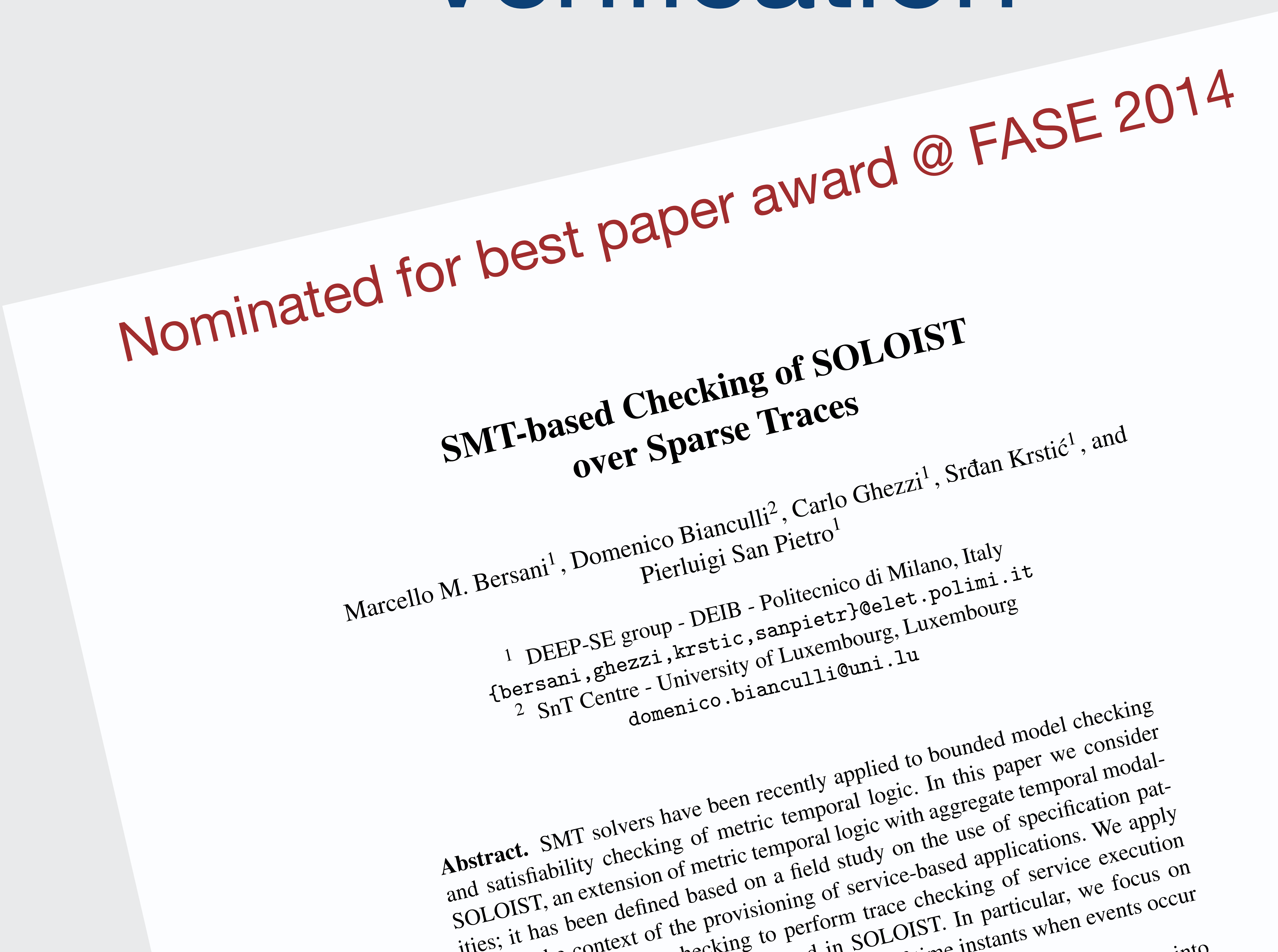
SMT solver

Verification

QF-EUFIDL



Verification



Nominated for best paper award @ FASE 2014

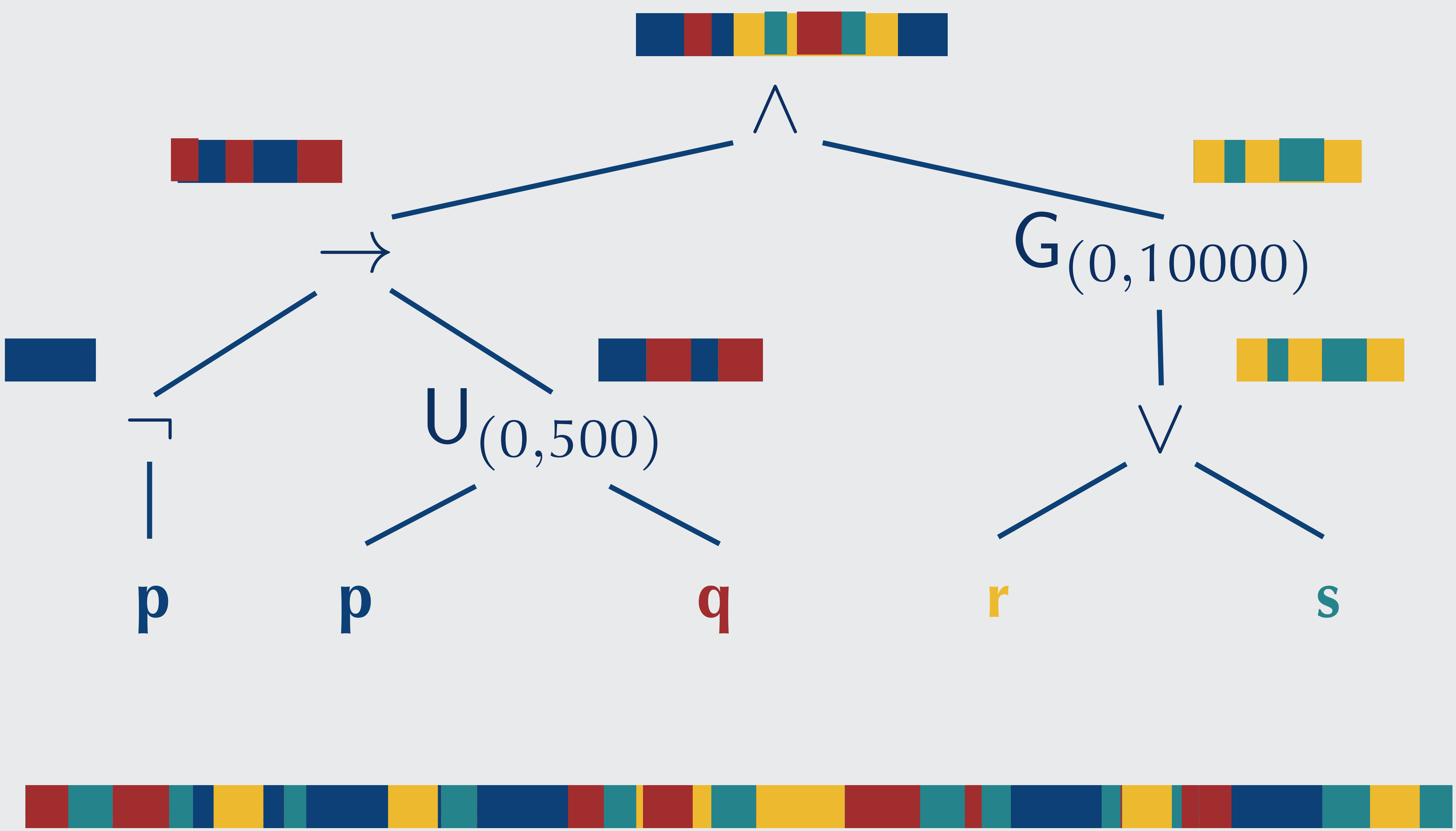
SMT-based Checking of SOLOIST over Sparse Traces

Marcello M. Bersani[1], Domenico Bianculli[2], Carlo Ghezzi[1], Srđan Krstić[1], and Pierluigi San Pietro[1]

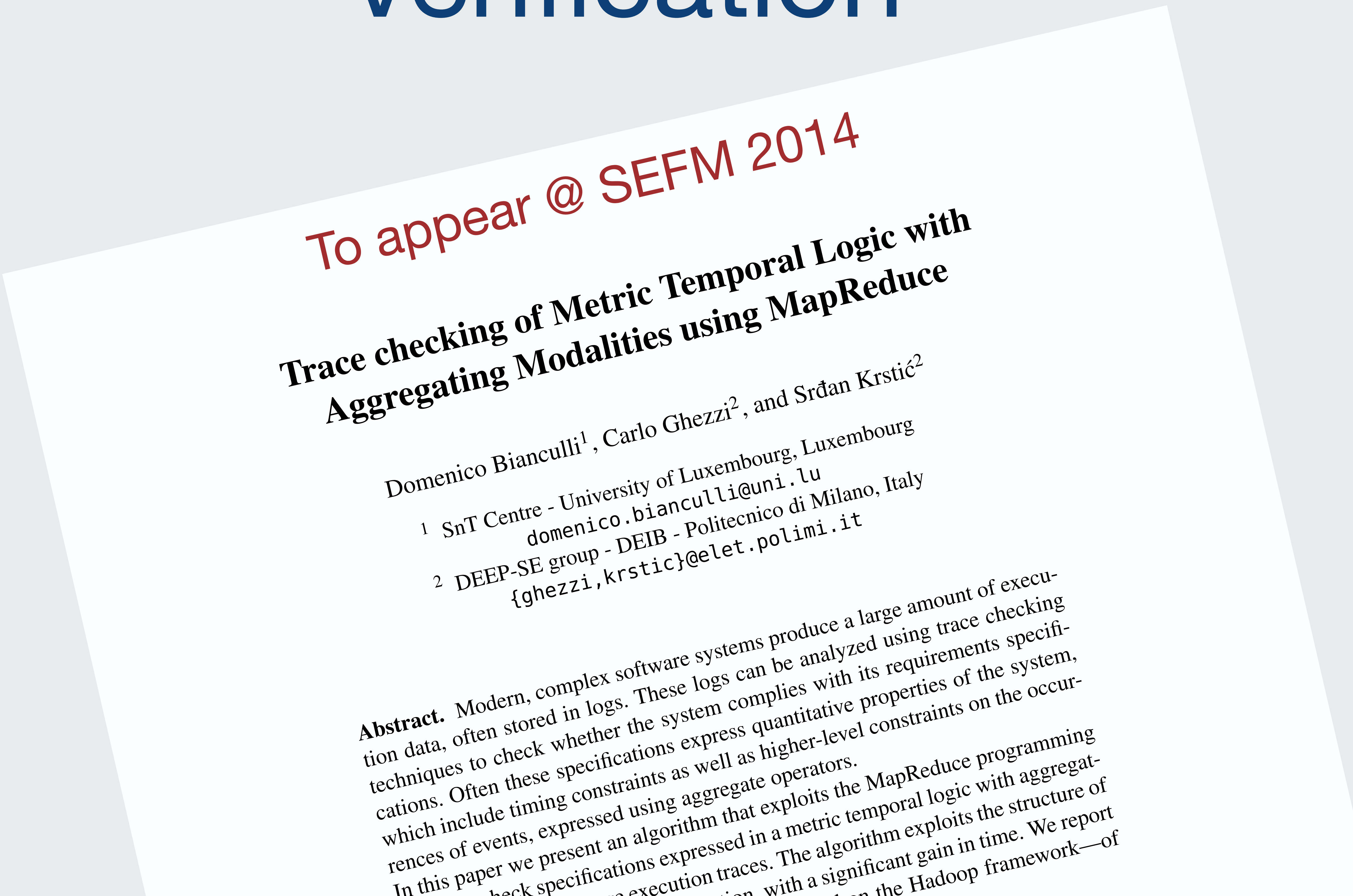
[1] DEEP-SE group - DEIB - Politecnico di Milano, Italy
`{bersani,ghezzi,krstic,sanpietr}@elet.polimi.it`
[2] SnT Centre - University of Luxembourg, Luxembourg
`domenico.bianculli@uni.lu`

Abstract. SMT solvers have been recently applied to bounded model checking and satisfiability checking of metric temporal logic. In this paper we consider SOLOIST, an extension of metric temporal logic with aggregate temporal modalities; it has been defined based on a field study on the use of specification patterns in the context of the provisioning of service-based applications. We apply ... checking to perform trace checking of service execution ... d in SOLOIST. In particular, we focus on ... time instants when events occur ... into

Verification



Verification



To appear @ SEFM 2014

Trace checking of Metric Temporal Logic with Aggregating Modalities using MapReduce

Domenico Bianculli[1], Carlo Ghezzi[2], and Srđan Krstić[2]

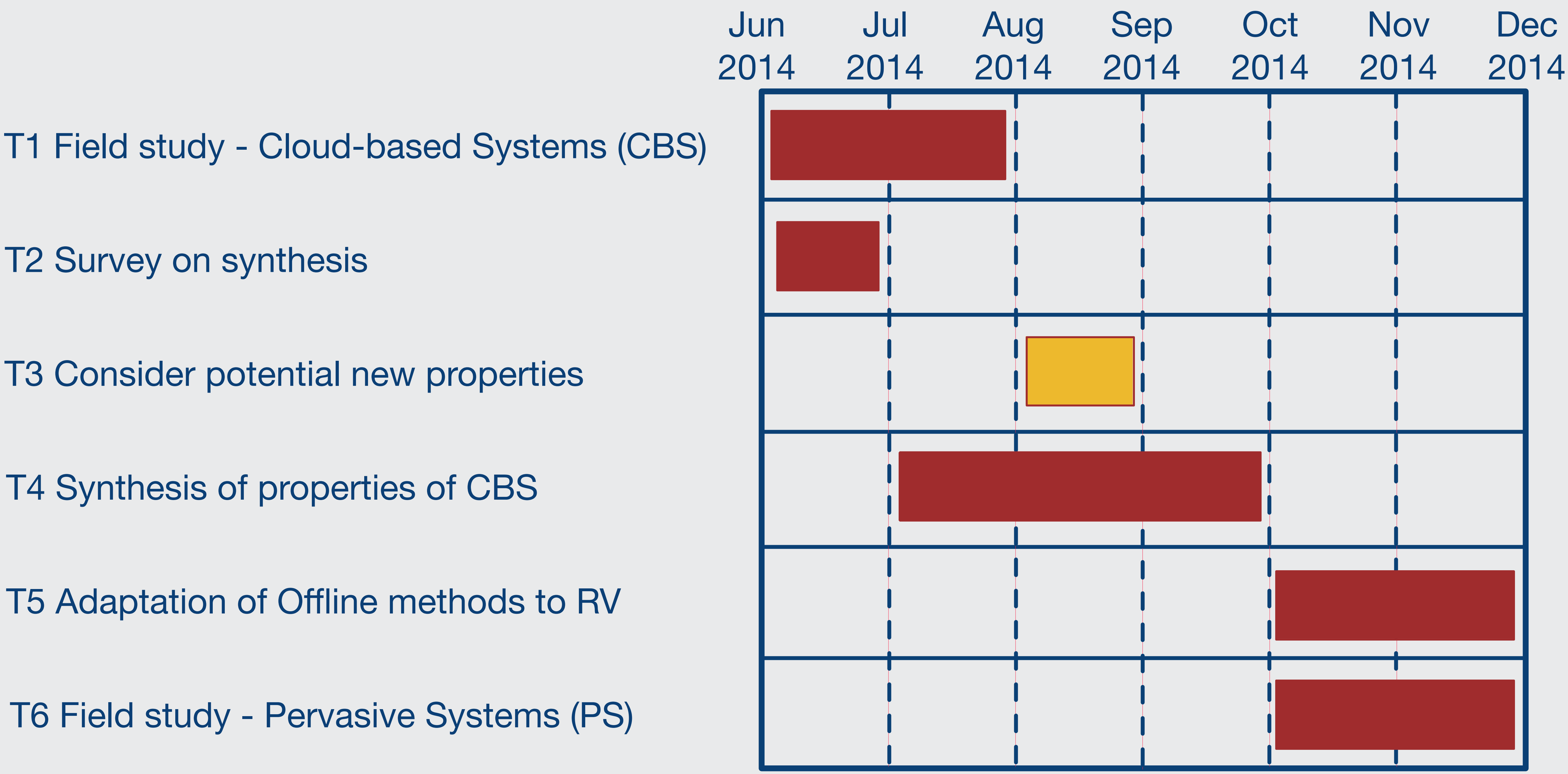
[1] SnT Centre - University of Luxembourg, Luxembourg
domenico.bianculli@uni.lu

[2] DEEP-SE group - DEIB - Politecnico di Milano, Italy
{ghezzi,krstic}@elet.polimi.it

Abstract. Modern, complex software systems produce a large amount of execution data, often stored in logs. These logs can be analyzed using trace checking techniques to check whether the system complies with its requirements specifications. Often these specifications express quantitative properties of the system, which include timing constraints as well as higher-level constraints on the occurrences of events, expressed using aggregate operators.

In this paper we present an algorithm that exploits the MapReduce programming ... check specifications expressed in a metric temporal logic with aggregat- ... execution traces. The algorithm exploits the structure of ... ion, with a significant gain in time. We report ... on the Hadoop framework—of

Future Schedule



Quantitative Properties: Specification, Verification and Synthesis

Srđan Krstić

advisor:
prof. Carlo Ghezzi

co-advisor:
Dr. Domenico Bianculli

Acknowledgments

prof. Carlo Ghezzi

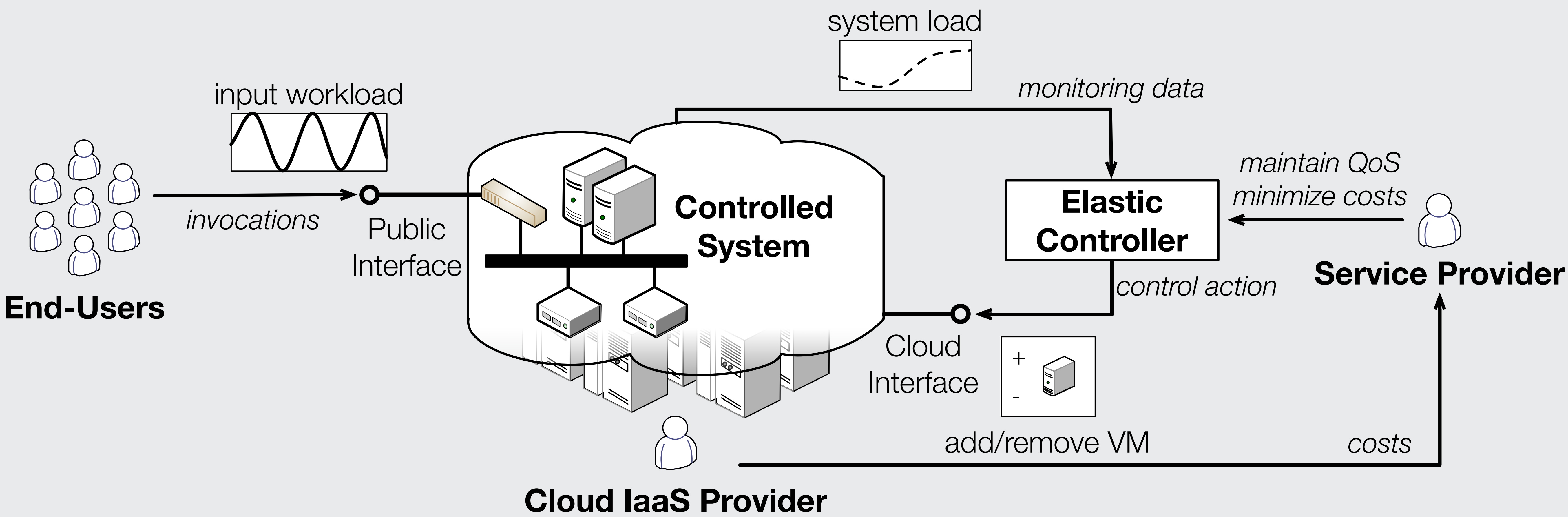
Dr. Domenico Bianculli

Additional Slides

Property Specification Patterns

Dwyer et al	Konrad et al	Gruhn et al	Bianculli et al
1 Absence	1 Minimum duration	1 Time bounded existence	1 Avg response time
2 Universality	2 Maximum duration	2 Time bounded response	2 Counting # events
3 Existence	3 Bounded recurrence	3 Precedence with delay	3 Avg # events
4 Bounded existence	4 Bounded response	4 Time restricted precedence	4 Max # events
5 Precedence	5 Bounded invariance		5 Absolute time
6 Response			6 Elapsed time
7 Response chains			7 Data-awareness
8 Precedence chains			
9 Constrained chain			

Cloud-Based Elastic System



Property Groups

Elasticity

Eagerness

Sensitivity

Plasticity

Resource Management

Precision

Oscillation

Resource thrashing

Cool-down period

Bounded concurrent adaptations

Bounded resource usage

Quality of Service

Bounded QoS degradation

Bounded actuation delay

Examples

Sequence of events

“For a period of 7 days, the application will successfully process a minimum of 500,000 customer orders per day”

Numerical bound

Aggregate transformation

Examples

Multiplicity of events

"The missile avionics system shall update the position of the ailerons exactly 20 times a second."

Numerical bound

Examples

“The university website shall not have more than 5 hours of scheduled downtime per month and not more than an average of 1 hour of unscheduled downtime per month.”

Related Work

Finkbeiner et al.

- collect statistics over run-time executions
- extend LTL to collect values
- language does not support timing information

Basin et al.

- MFOTL with aggregates
- more aggregates than SOLOIST
- values of the relation parameters vs occurrences

Bauer et al.

- PTLTLFO - past time linear temporal logic with first-order (guarded) quantifiers and counting modality
- lacks timing information and bounded windows

Barre et al.

- MapReduce based approach
- plain LTL (without timing information and aggregates)
- inefficient handling of tuples (no sorting)
- no multi-operand conjunction and disjunction