A Dynamic Risk-based Access Control Architecture for Cloud Computing

Daniel R. dos Santos, Carla M. Westphall, Carlos B. Westphall
{danielrs,carlamw,westphal}@inf.ufsc.br

Networks and Management Laboratory
Federal University of Santa Catarina
Florianópolis, SC - Brazil

Agenda

- Introduction
- Related work
- Risk-based access control
- Proposed architecture
- Implementation and experiments
- Conclusion and future work

Introduction

- Cloud computing is a successful paradigm and cloud federations aim to make it even more efficient and scalable by sharing resources among providers

- In highly distributed, dynamic and heterogeneous environments, traditional access control models present problems, such as: scalability, flexibility and the use of static policies

- Dynamic access control models, like risk-based, provide greater flexibility and are able to handle exceptional requests (“break the glass”)

Introduction

- We present a model for dynamic risk-based access control for cloud computing

- The system uses quantification and aggregation of risk metrics that are defined in risk policies, which are created by the owners of the cloud resources

- It is built on top on an XACML architecture and allows the use of ABAC coupled with risk analysis

Related Work

- Fall et al. [1] - presents the first idea of risk-based AC for cloud. Propose using NSA RAdAC, but show no implementation

- Arias-Cabarcos et al. [2] - proposes the use of a fixed set of risk metrics for establishing identity federations in the cloud

- Sharma et al. [3] - uses risk-based AC on top of RBAC for cloud e-Health. Their model has 3 metrics (Confidentiality, Integrity and Availability)

Risk-based Access Control

- Traditional access control models employ static authorization, i.e., every decision is pre-established, based on the policies

- The idea behind dynamic access control is that the access requests must be analyzed taking into account contextual and environmental information such as security risk, operational need, benefit and others

- Real applications may require the violation of security policies, and the support for exceptional access requests is known as “break the glass”

Risk-based Access Control

- Uses a function that evaluates in “real time” each request

- Risk analysis can be qualitative, with levels of risk, or quantitative, where risk is usually defined as: Probability X Impact

- Many approaches to risk quantification: fuzzy logic, machine learning, probabilistic inference, …
  - usually based on the history of users and access

Proposed Architecture

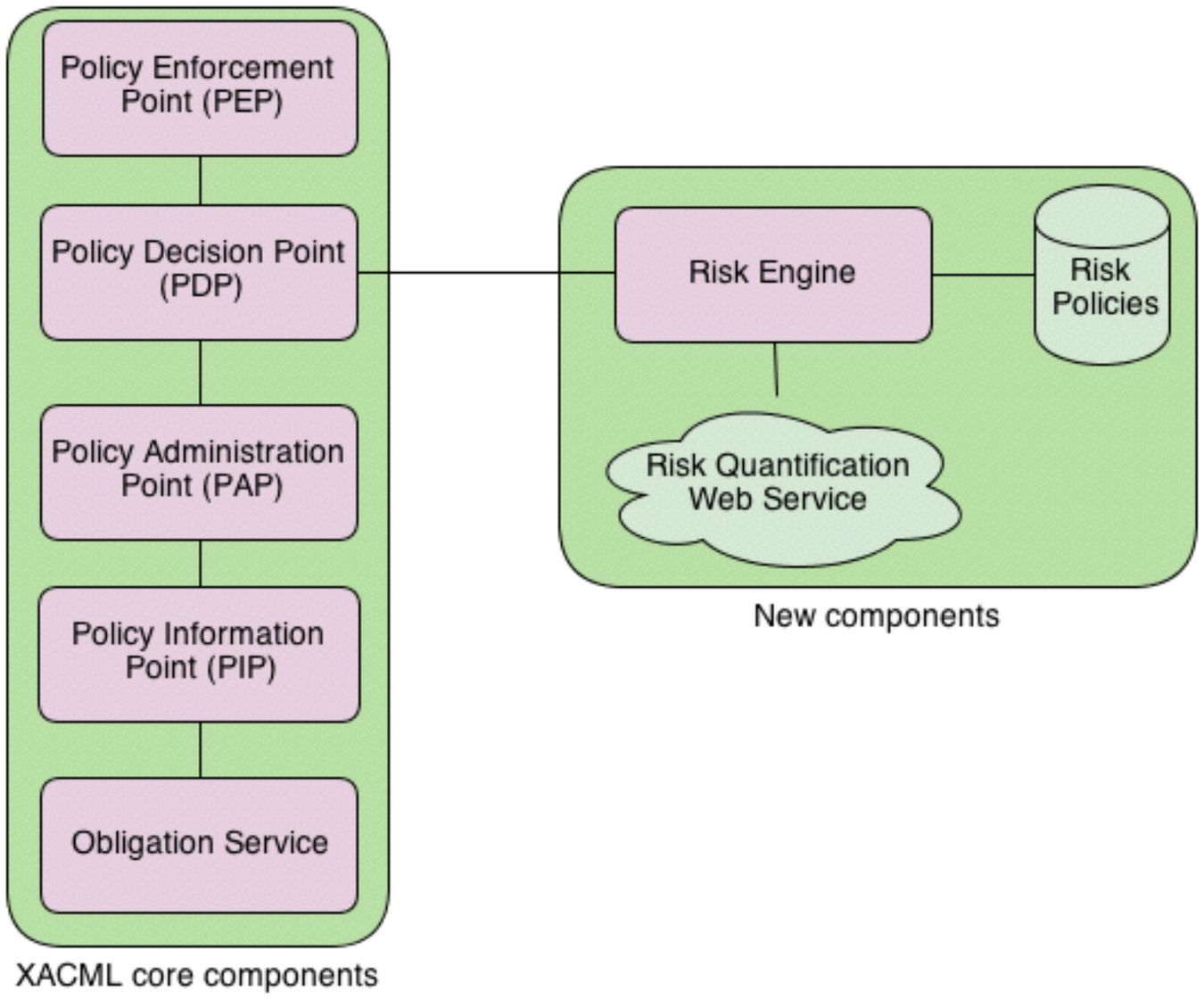
- XACML extension. ABAC and risk-based are taken in parallel and then combined to reach a final decision.
  - Combination rules: Deny overrides, Permit overrides, ABAC precedence, Risk precedence

- Risk decision is based on XML risk policies associated to a resource. A policy defines a set of risk metrics, how to quantify and aggregate them and an acceptable risk threshold

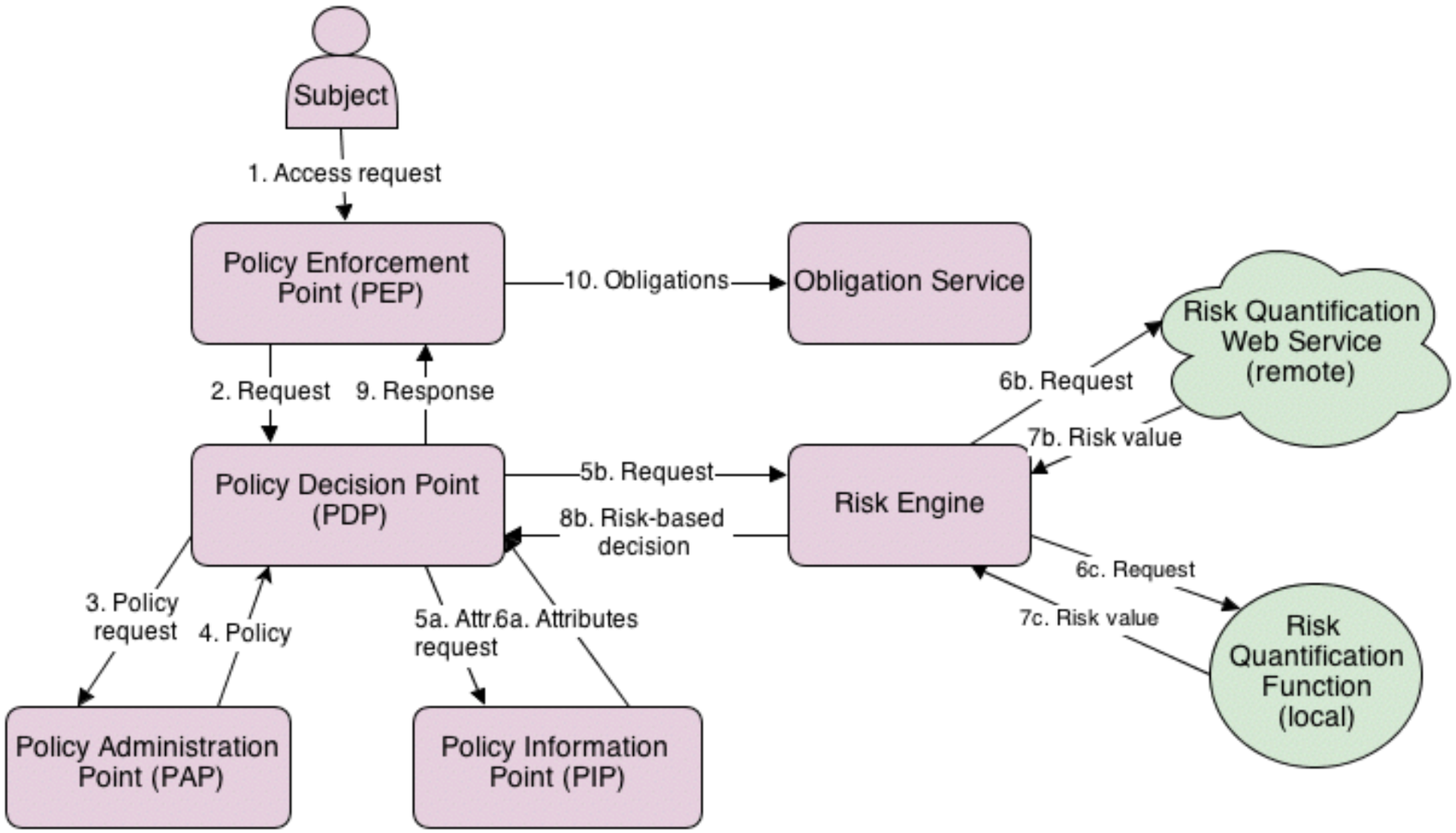
- Quantification and aggregation methods can be local (in the CSP) or external, defined by the resource owner as a web service

- The CSP has a basic risk policy, defining the maximum risk level accepted by it

Overview


Policy Enforcement Point (PEP)
Policy Decision Point (PDP)
Policy Administration Point (PAP)
Policy Information Point (PIP)
Obligation Service
XACML core components
Risk Engine
Risk Policies
Risk Quantification Web Service
New components

Decision process



Case study - cloud federation

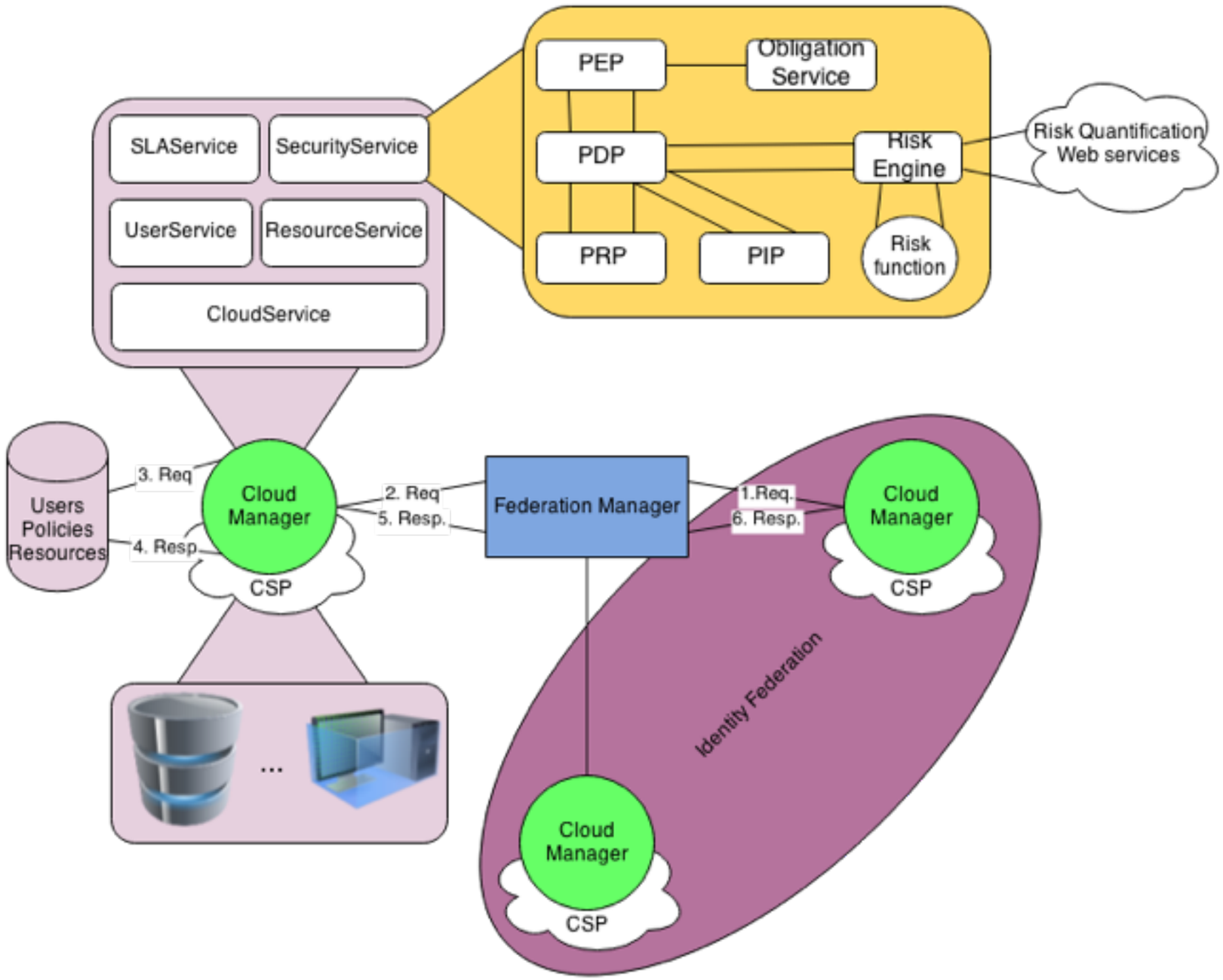
- Identity and Access Management is a big challenge when setting up a cloud federation

- It involves a notion of trust, which is usually mediated by an identity federation, this has two major issues:
  - trust agreements and interoperability

- To decrease the level of trust needed among participating clouds, we incorporate the notion of risk

- Also, interoperability may be increased, because a missing attribute in a message may also be considered as a risk factor, instead of stopping communication

Case study - cloud federation



Considerations

- The architecture allows a flexible AC system

- Risk analysis may be too subjective
  - The support of Obligations is essential

- Risk policies allow the use of many risk metrics, using diverse quantification and aggregation methods from different sources

- The main limitation is the performance overhead due to the processing of the risk policies and the quantification of the risk metrics

Implementation

- Three stages:
  - Access control architecture; Cloud federation; Risk quantification and aggregation methods

- Python, ndg-xacml, ZeroMQ, web.py, peewee, MySQL, OpenNebula

- Two risk policies implemented for tests:
  - Sharma et al. [3] : ((a * p1) + (i * p2) + (c * p3) + pastScore)

  - Britton and Brown [4] : 27 metrics

Experiments - risk policy

```
<rp:risk-policy version="1.0" xmlns:rp="http://inf. ufsc.br/~danielrs">
<rp:resource id="1"/><rp:user id="1"/> <rp:metric-set name="sharma2012">
        <rp:metric>
            <rp:name>Confidentiality</rp:name>
            <rp:quantification>https://localhost:8443/quantify-conf</rp:quantification>
        </rp:metric>
        <rp:metric>
            <rp:name>Availability</rp:name>
            <rp:quantification>https://localhost:8443/quantify-avail</rp:quantification>
</rp:metric>
        <rp:metric>
            <rp:name>Integrity</rp:name>
            <rp:quantification>https://localhost:8443/quantify-int</rp:quantification>
        </rp:metric>
    </rp:metric-set>
    <rp:aggregation-engine>https://localhost:8443/aggregate</rp:aggregation-engine>
    <rp:risk-threshold>1.5</rp:risk-threshold>
</rp:risk-policy>
```

Experiments

TABLE I. PERFORMANCE OF RISK POLICIES

Policy	min. (ms)	max. (ms)	avg (ms)
XACML	0.925	4.278	1.040
XACML+[34]	1.986	11.973	2.436
XACML+[33]	4.395	14.234	5.352

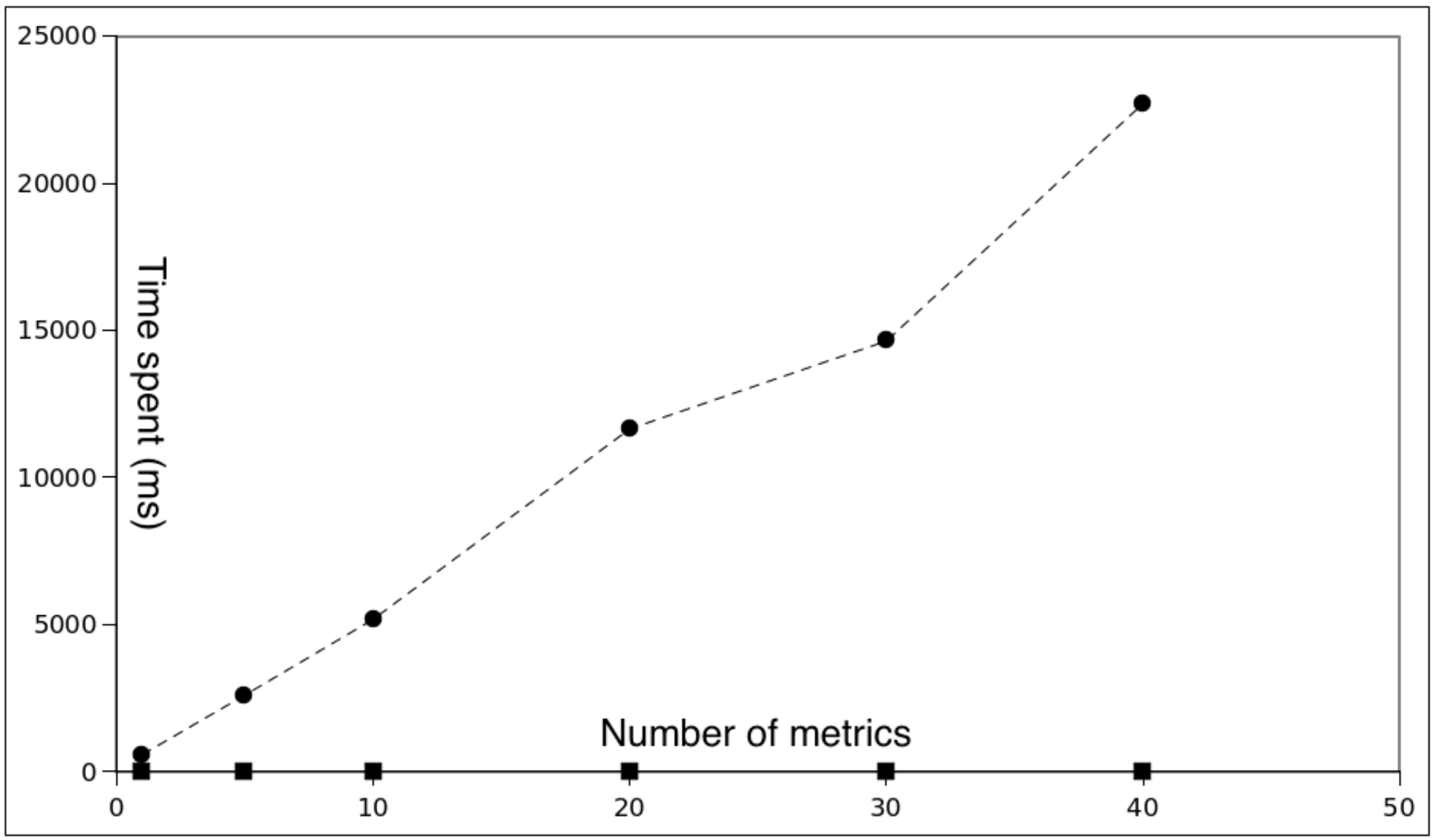
TABLE II. PERFORMANCE WITH A VARYING NUMBER OF METRICS

Number of metrics	min. (ms)	max. (ms)	avg (ms)
1	1.832	12.130	2.243
10	2.612	12.876	3.171
100	10.922	60.442	14.030
1000	96.041	175.245	121.383
10000	1168.511	1517.364	1361.025

TABLE III. PERFORMANCE WITH LOCAL AND EXTERNAL METRICS

Case	min. (ms)	max. (ms)	avg (ms)
A	1.057	9.372	1.46
B	1.824	15.564	4.574
C	1556.182	2813.56	1726.71
D	3247.563	10350.5	4220.6

Experiments



Conclusion

- AC systems for the cloud are of great importance and traditional AC models are not enough for the cloud

- Risk-based AC tend to be very specific to a given scenario, we tried to make it more general, to be applied in a CSP

- We presented, implemented and evaluated the performance of our architecture

- As future work, we would like to: integrate the architecture into a mature cloud federation project; implement other risk quantification methods; improve the performance of external metrics (caching, concurrent requests, …); and develop a reference set of risk metrics for the cloud

Thank you!

Daniel R. dos Santos, Carla M. Westphall, Carlos B. Westphall
{danielrs,carlamw,westphal}@inf.ufsc.br

References

- [1] D. Fall, G. Blanc, T. Okuda, Y. Kadobayashi, and S. Yamaguchi, “Toward Quantified Risk-Adaptive Access Control for Multi-tenant Cloud Computing,” in *Proceedings of the 6th Joint Workshop on Information Security*, October 2011.

- [2] P. Arias-Cabarcos, F. Almenárez-Mendoza, A. Marín- López, D. Díaz-Sánchez, and R. Sánchez-Guerrero, “A metric-based approach to assess risk for “on cloud” federated identity management,” *Journal of Network and Systems Management*, vol. 20, pp. 513–533, 2012.

- [3] M. Sharma, Y. Bai, S. Chung, and L. Dai, “Using risk in access control for cloud-assisted ehealth,” in *IEEE 14th International Conference on High Performance Comput- ing and Communication, 2012*, 2012, pp. 1047–1052.

- [4] D. Britton and I. Brown, A security risk measurement for the RAdAC model, 2007.