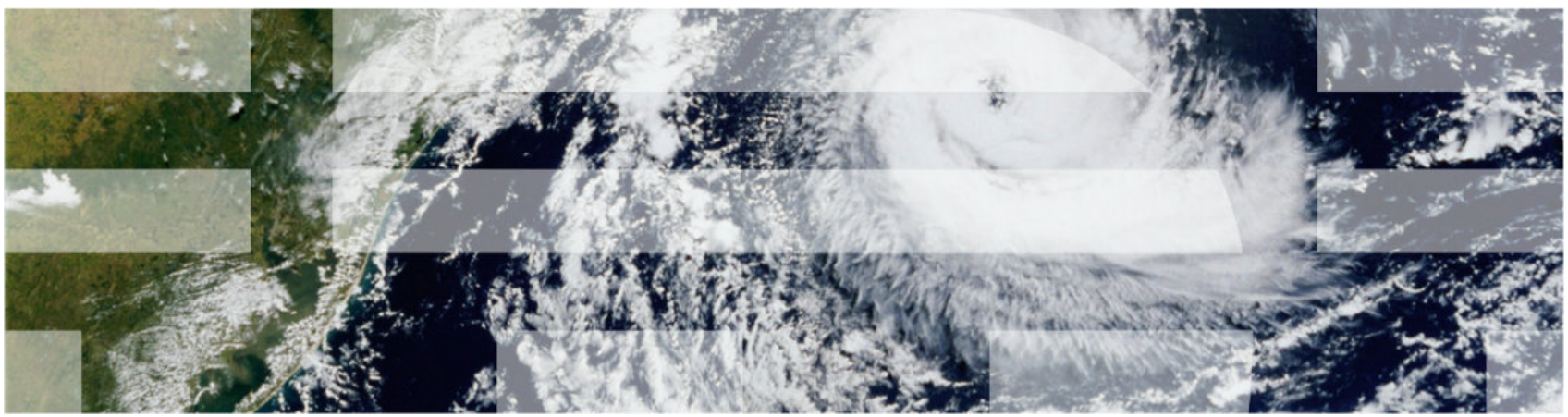
WISE-SPL: Bringing Multi-tenancy to the Weather InSights Environment System

V. Segura, Leonardo P. Tizzei, J. Ramirez, M. dos Santos, L. Azevedo, R. Cerqueira

May 19, 2015



Agenda

1 Introduction

2 Applying SPL techniques

3 Discussion: benefits and limitations

4 Conclusion and Future Work

Agenda

1 Introduction

2 Applying SPL techniques

3 Discussion: benefits and limitations

4 Conclusion and Future Work

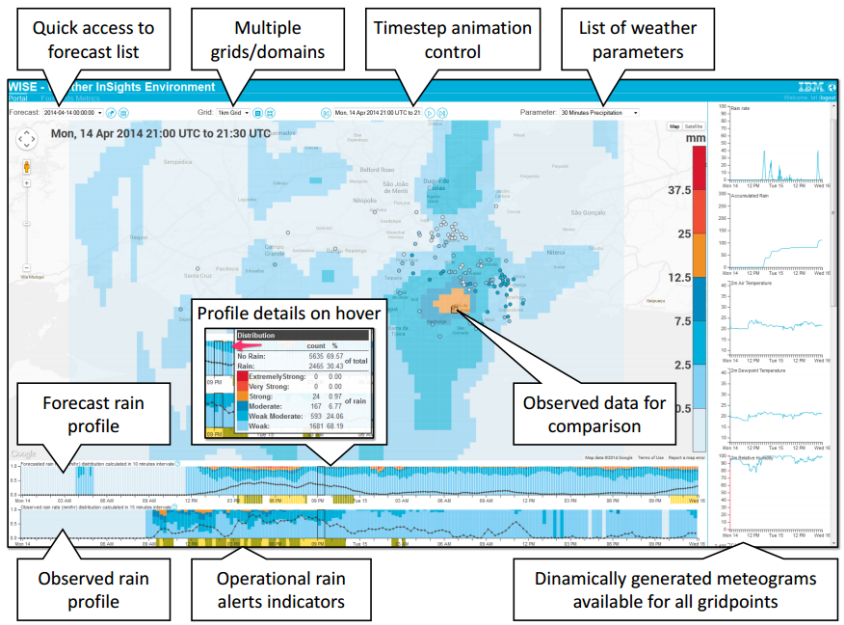
Introduction

- City Operation Centers around the world tackle many important challenges during their routine job.
- Weather conditions affect operations of many cities and companies.

Figure: Acre, Brazil - March, 2015

Weather InSights Environment (WISE) (1/3)

- The WISE system serves as a central place to gather and present weather related information.
- Visualization platform
- Combination of forecast data generated by numerical weather prediction systems with the observed data obtained from sensors network $\Rightarrow$ the visual correlation between both data.



Weather InSights Environment (WISE) (2/3)

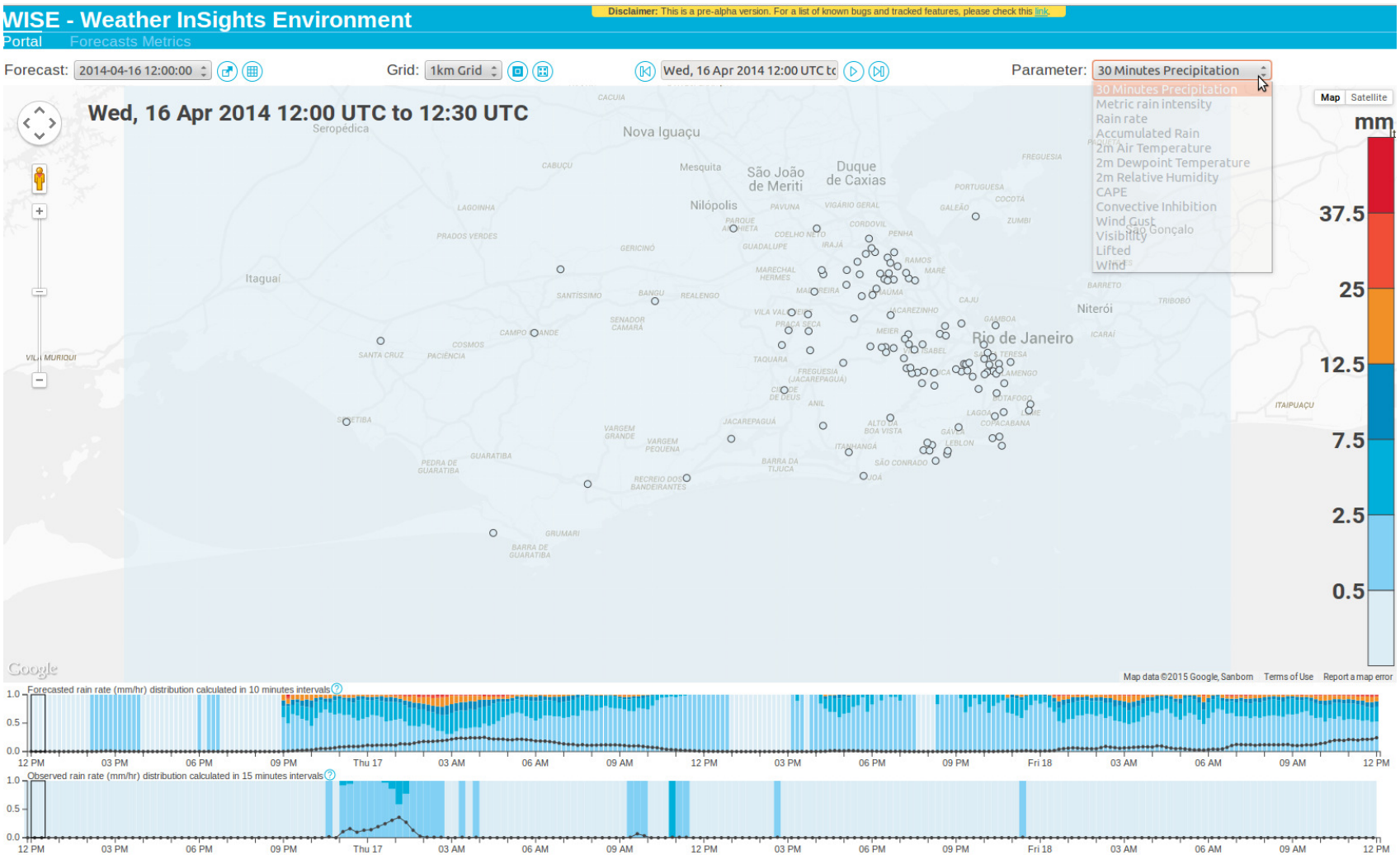
BRAZIL

CHINA

CANADA

	Brazil	**China**	**Canada**
base map	GoogleMaps	ArcGIS	GoogleMaps
environmental properties	Air temperature, Rain, ...	Visibility, Air pollution	Air temperature, Rain, ...
rain verification	✓	x	x

Weather InSights Environment (WISE) (3/3)



Moving WISE to the cloud

Cloud Concepts

- Cloud computing enables on-demand access to a shared pool of computing resources [Mell and Grance 2011].
- Multi-tenant cloud computing tools support multiple clients by enabling customizations that fit client requirements [Rimal et al. 2011].

Technical and Non-Technical aspects

- Facility to obtain computing resources
- Company's strategy
- Compliance with clients' requirements
- Developers will learn new technologies

Problem and Contribution

Problem

- To what extent SPL techniques support the development of a multi-tenant WISE?

Contribution

- Application of existing SPL techniques to build a multi-tenant application.
- From 3 existing legacy applications, we modeled a SPL based on extractive approach
- Identification of benefits and limitations of existing techniques.

Agenda

1 Introduction

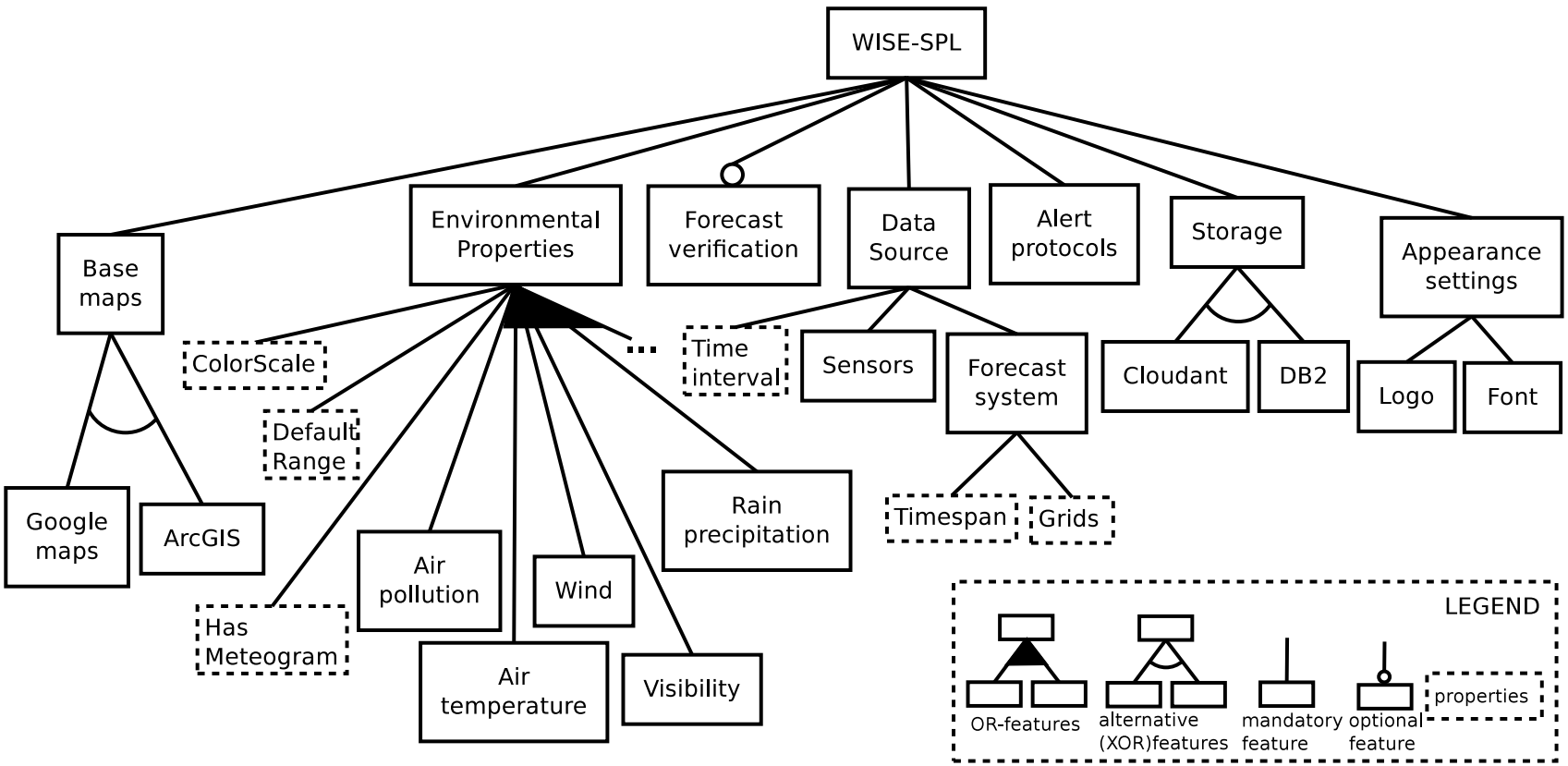
2 Applying SPL techniques

3 Discussion: benefits and limitations

4 Conclusion and Future Work

Feature Modeling

- Feature modeling = commonalities and variabilities among tenants' requirements



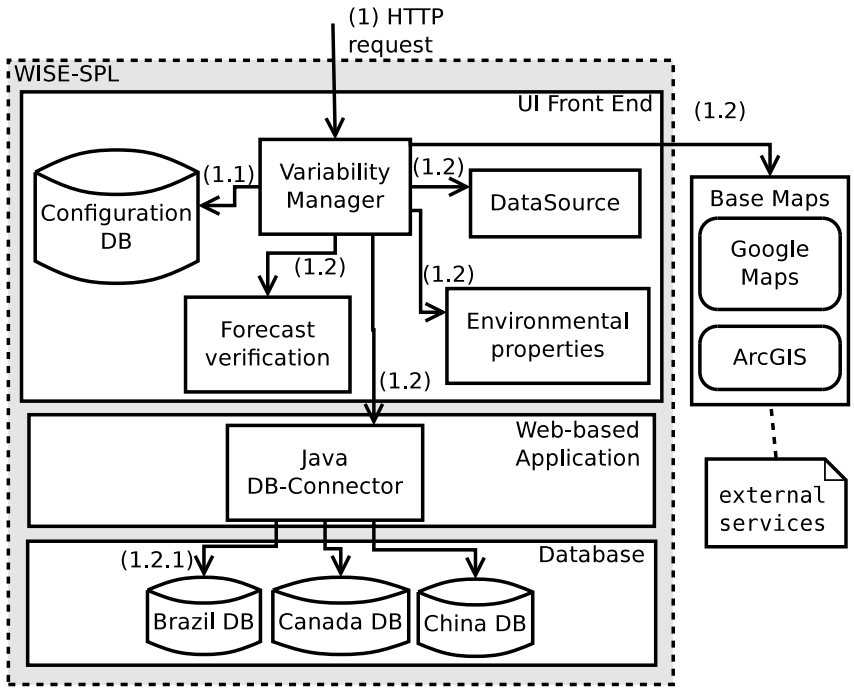
WISE-SPL Configuration

- SPLOT tool [Mendonça et al. 2009]
- Feature modeling, automated analysis, product configuration $\Rightarrow$ generates CSVs or text files $\Rightarrow$ runtime variability

```
WISE                        templateModel 1 propagated 1
Environmental properties    mandatory     1 propagated 1
Air pollution               grouped       0 manual     8
Visibility                  grouped       0 manual     9
Wind                        grouped       1 manual     4
Air temperature             grouped       1 manual     2
Rain precipitation          grouped       1 manual     3
...
```

WISE-SPL Architectural Design

- Feature-Architecture Mapping (FArM) method [Sochos et al. 2006]
- Mapping features to architectural elements (i.e., components)
- Minimization of feature tangling and scattering



Agenda

1 Introduction

2 Applying SPL techniques

3 Discussion: benefits and limitations

4 Conclusion and Future Work

Discussion

- Feature model contributed to identify common and variable features of different tenants.
- Few SPL approaches detailed variability **implementation** for web-based SPLs
- *“..when implementing SPLs (...), such as Web systems, we notice a gap between theory and practice.”* [Machado et al. 2014]

Cloud Architecture Variability Modeling

- Elasticity and disaster recovery are common features of cloud applications.
- [Hallsteinsen et al. 2006] applied SPL techniques to build adaptive systems.
- [Nascimento et al. 2013] used Common Variability Language to support fault-tolerant service compositions.
- But few works apply SPL techniques to address elasticity or disaster recovery of cloud computing architectures.

Agenda

1 Introduction

2 Applying SPL techniques

3 Discussion: benefits and limitations

4 Conclusion and Future Work

Conclusion and Future Work

Conclusion

- Building WISE-SPL cloud application via extractive approach
- Identification of benefits and limitations of SPL techniques

Future Work

- Ongoing implementation of WISE-SPL
- Rigorous study to assess reuse
- Assessment of other techniques, e.g., TOSCA [Brogi et al. 2014]

Thanks

Thanks for your attention!
Questions?

e-mail:
ltizzei@br.ibm.com


Brogi, A., Soldani, J., and Wang, P. (2014).
TOSCA in a Nutshell: Promises and Perspectives.
In Villari, M., Zimmermann, W., and Lau, K.-K., editors, *Service-Oriented and Cloud Computing*, volume 8745 of *Lecture Notes in Computer Science*, page 171–186. Springer Berlin Heidelberg.


Hallsteinsen, S., Stav, E., Solberg, A., and Floch, J. (2006).
Using product line techniques to build adaptive systems.
In *Software Product Line Conference, 2006 10th International*, pages 10–pp. IEEE.


Machado, I. d. C., Santos, A. R., Cavalcanti, Y. C., Trzan, E. G., de Souza, M. M., and de Almeida, E. S. (2014).
Low-level variability support for web-based software product lines.
In *Proceedings of the Eighth International Workshop on Variability Modelling of Software-Intensive Systems*, page 15. ACM.


Mell, P. and Grance, T. (2011).
The NIST Definition of Cloud Computing.
Technical report, NIST.


Mendonça, M., Branco, M., and Cowan, D. (2009).
SPLOT: software product lines online tools.
In *Proceedings of the ACM SIGPLAN conference companion on Object oriented programming systems languages and applications*, pages 761–762. ACM.


Nascimento, A., Rubira, C., and Castor, F. (2013).
Using CVL to support self-adaptation of fault-tolerant service compositions.
In *Proceedings of the International Conference on Self-Adaptive and Self-Organizing Systems*, pages 261–262.


Rimal, B., Jukan, A., Katsaros, D., and Goeleven, Y. (2011).
Architectural Requirements for Cloud Computing Systems: An Enterprise Cloud Approach.
Journal of Grid Computing, 9(1):3–26.


Sochos, P., Riebisch, M., and Philippow, I. (2006).
The feature-architecture mapping (farm) method for feature-oriented development of software product lines.
In *ECBS '06: Proceedings of the 13th Annual IEEE International Symposium and Workshop on Engineering of Computer Based Systems*, pages 308–318, Washington, DC, USA. IEEE Computer Society.