

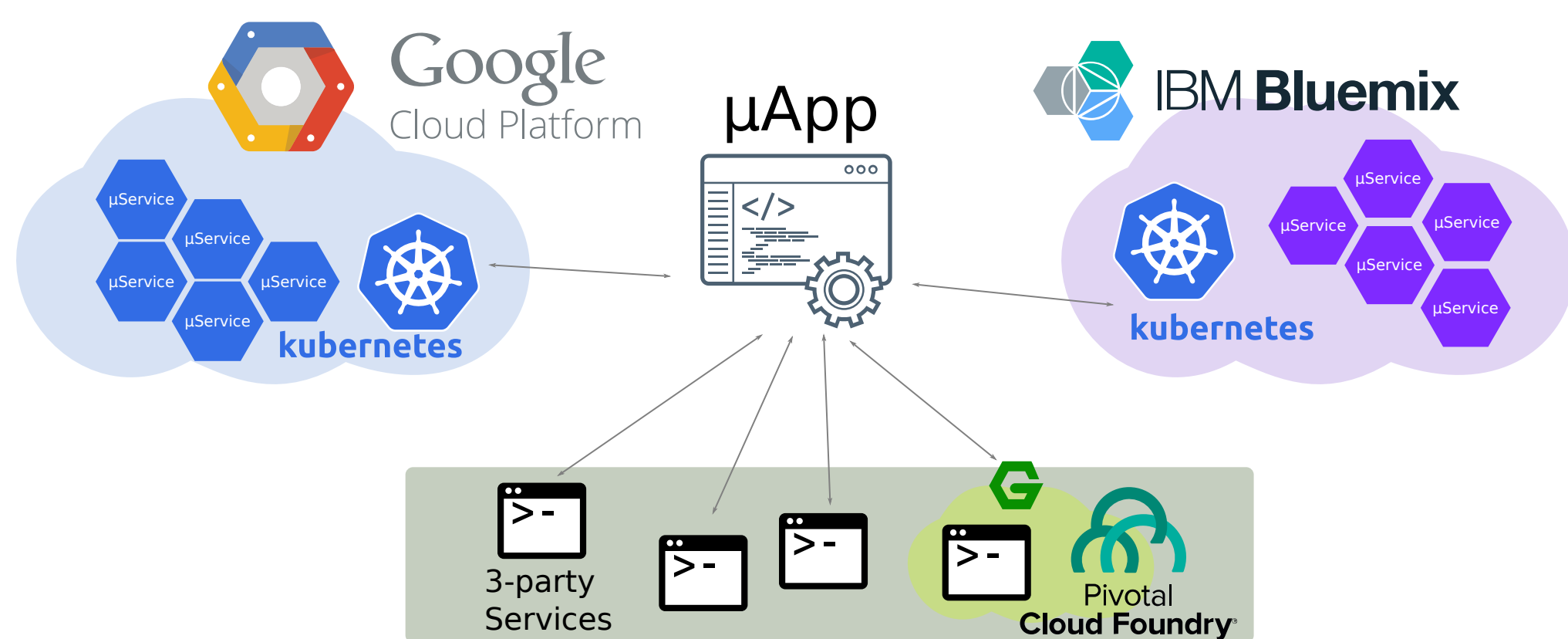
SUPPORTING MICROSERVICE EVOLUTION

Adalberto R. Sampaio Jr*, Harshavardhan Kadiyala§, Bo Hu§, John Steinbacher†, Tony Erwin‡, Nelson Rosa*, Ivan Beschastnikh§, Julia Rubin§

*University Federal of Pernambuco, §University of British Columbia, †IBM Canada and ‡IBM USA



Microservice-Based Apps (μ Apps)

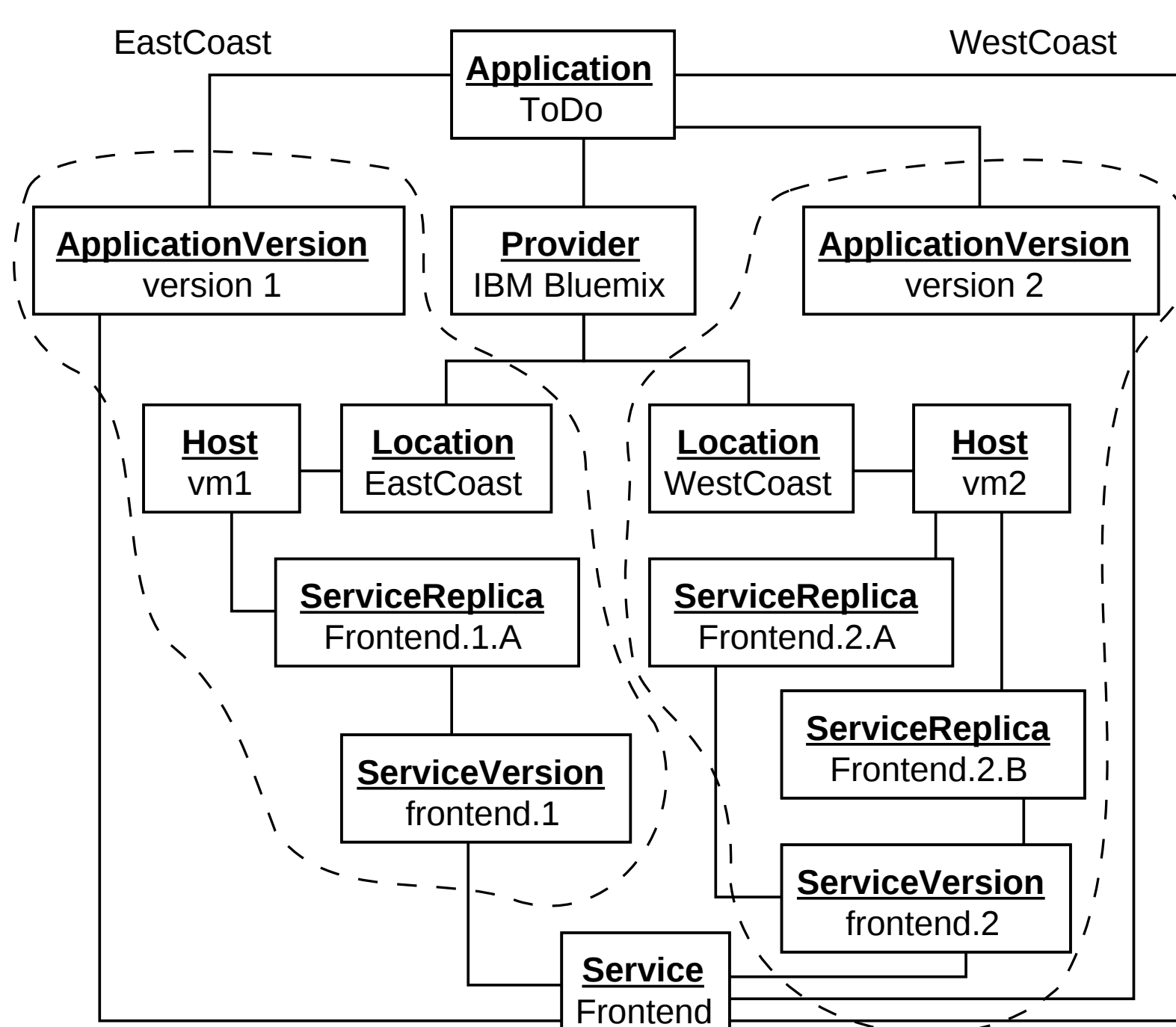


- Fast and easy to deploy
- Multilingual and multi-technology
- Loose dependencies \Rightarrow Constant evolution

Our approach: build an evolutionary model of the μ App

- Unify semantics of different data sources
- Extract and aggregate runtime information
- Use collected data to continuously model the μ App (Models@Run.time)
- Perform analysis on sequence of μ App models

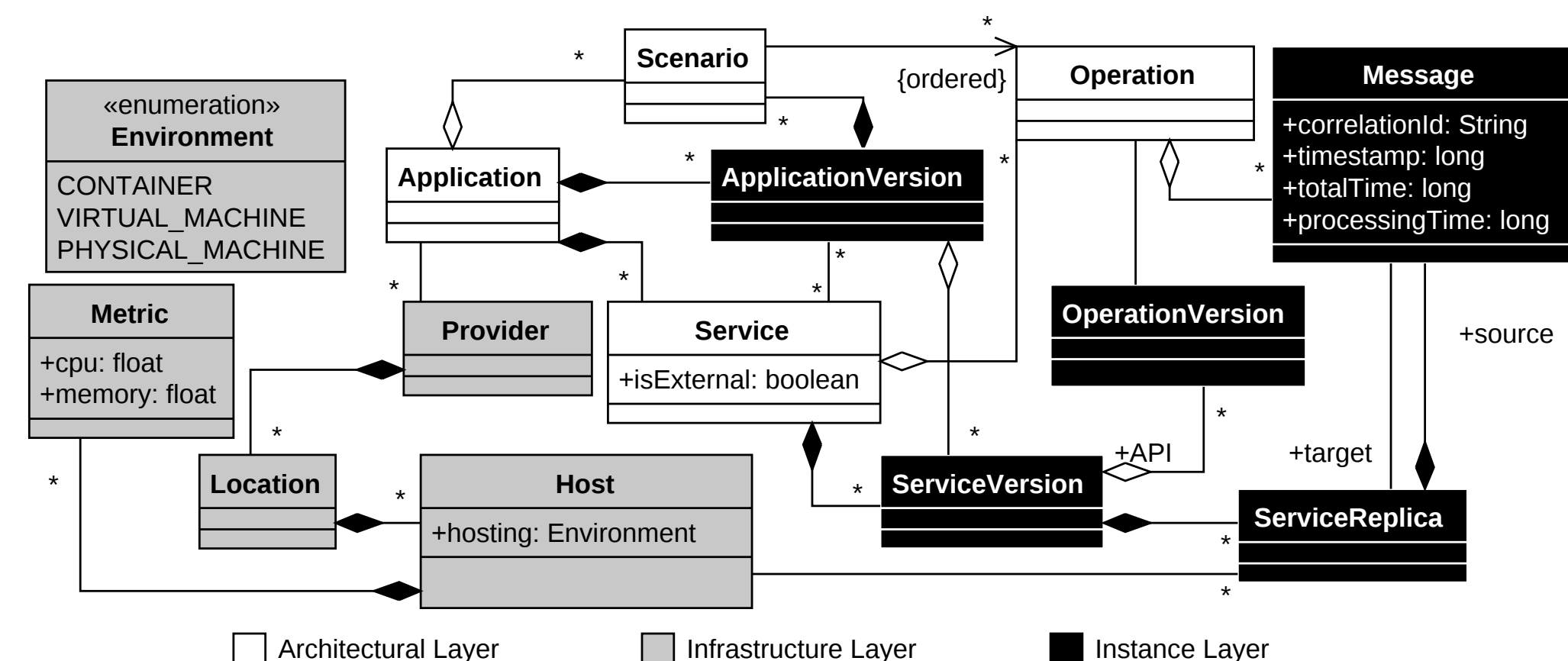
Example Model Instance



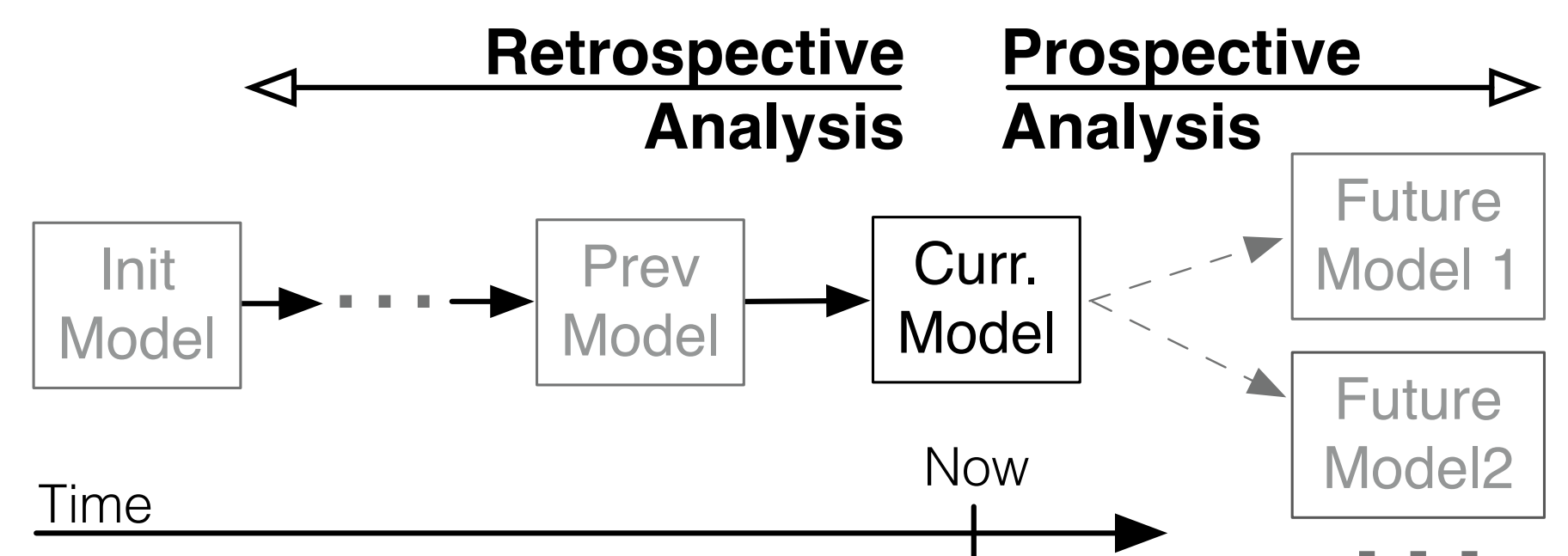
Challenges of Evolving μ Apps

- Upgrades brake inter-services compatibility
- Tracking down failure root causes complicated
- Many architectural choices; unclear which ones are the most appropriate
- Deployment configuration also needs to evolve

Proposed Evolutionary Model



Model Analysis



- Sequence of models over time allows for rich analyses
- **Retrospective example:** Study inter-services messages to recommend service refactorings
- **Prospective example:** Explore and instantiate new deployment configurations to optimize resources usage