



Institute for Software Integrated Systems

Vanderbilt University

---

# Investigating Dynamic Resource Management Solutions for Cloud Infrastructures Using Chameleon Cloud

Yogesh Barve, Anirban Bhattacharjee, Shweta Khare,  
Zhuangwei Kang, Robert Canady, Aniruddha Gokhale

**Presenter: Zhuangwei Kang**

Vanderbilt University





# Overview

## Themes:

- Interference-aware Performance Modeling--FECBench[1]
- Geo-distributed Latency-Aware Stream Processing -- [2]
- Resource Management for Emerging Workloads -- Distributed Co-simulation[3]/Machine Learning
- DevOps Cloud Automation & Orchestration -- CloudCAMP[4]

## Projects:

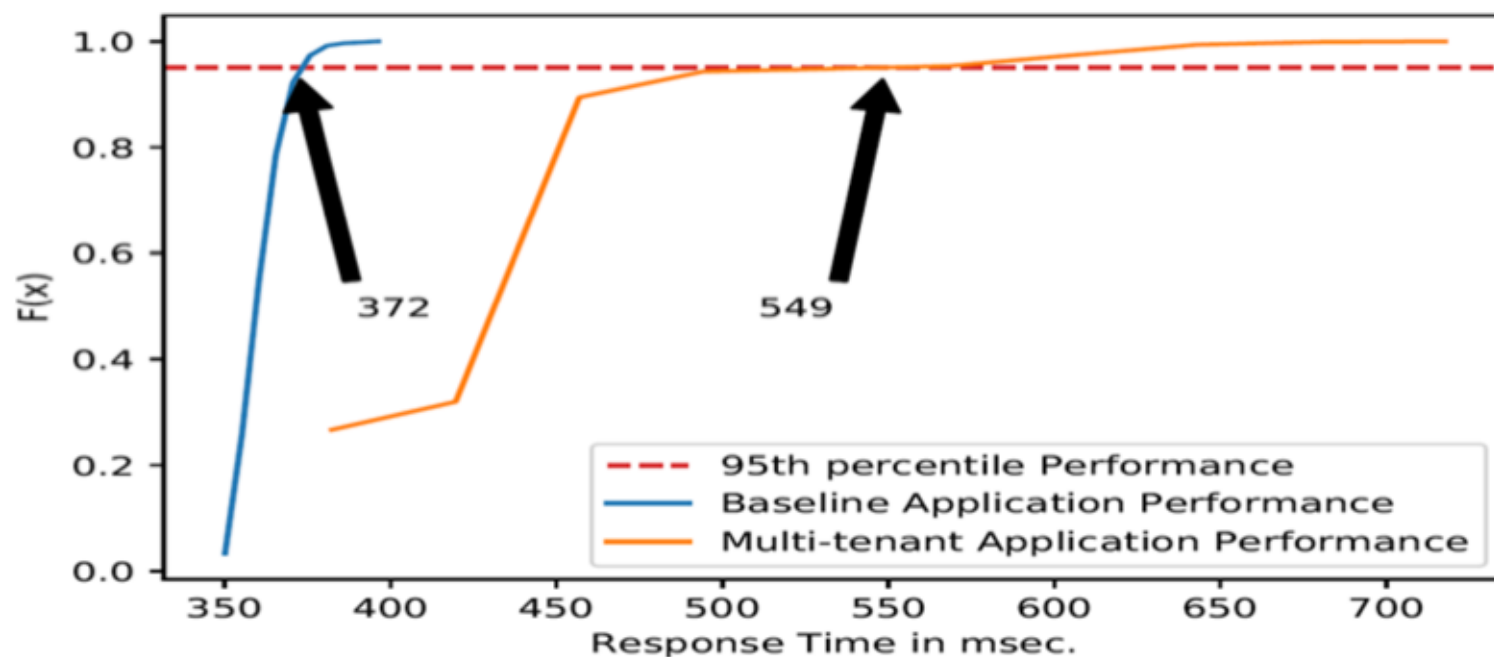
- [1] Barve, Yogesh, et al. "FECBench: An Extensible Framework for Pinpointing Sources of Performance Interference in the Cloud-Edge Resource Spectrum." 2018 IEEE/ACM Symposium on Edge Computing (SEC). IEEE, 2018.
- [2] Khare, Shweta et al. "Scalable Edge Computing for Low Latency Data Dissemination in Topic-Based Publish/Subscribe." 2018 IEEE/ACM Symposium on Edge Computing (SEC)(2018): 214-227.
- [3] Zhuangwei, Kang & Barve, Yogesh et al. "RESTfulSwarm: A Platform for Group Scheduling Distributed Co-Simulation Based on Docker Swarm."(2018)
- [4] Bhattacharjee, Anirban, et al. "A Model-Driven Approach to Automate the Deployment and Management of Cloud Services." 2018 IEEE/ACM International Conference on Utility and Cloud Computing Companion (UCC Companion). IEEE, 2018.





# Interference-aware Performance Modeling

## FECBench Motivation:



*Machine Learning Inception ResNet V2 Model Inference Serving Performance*



## Goals & Challenges for Performance Profiling

### ❖ **Goal 1: Benchmarking with Ease**

- Disparate set of applications for metrics collection
- Accidental complexities involved in manual writing of configuration scripts
- Lack of an integrated tool suite for configuration, monitoring and benchmarking

### ❖ **Goal 2: Automated Construction of Resource Stressors**

- Lack of tunable stressors that can stress resources across multiple dimensions (e.g., CPU 10%, memory Bandwidth 10%, Disk IO : 50%)
- Manual creation of stressor programs for different architectures/ platforms.

### ❖ **Goal 3: Minimize Prohibitive Profiling Cost**

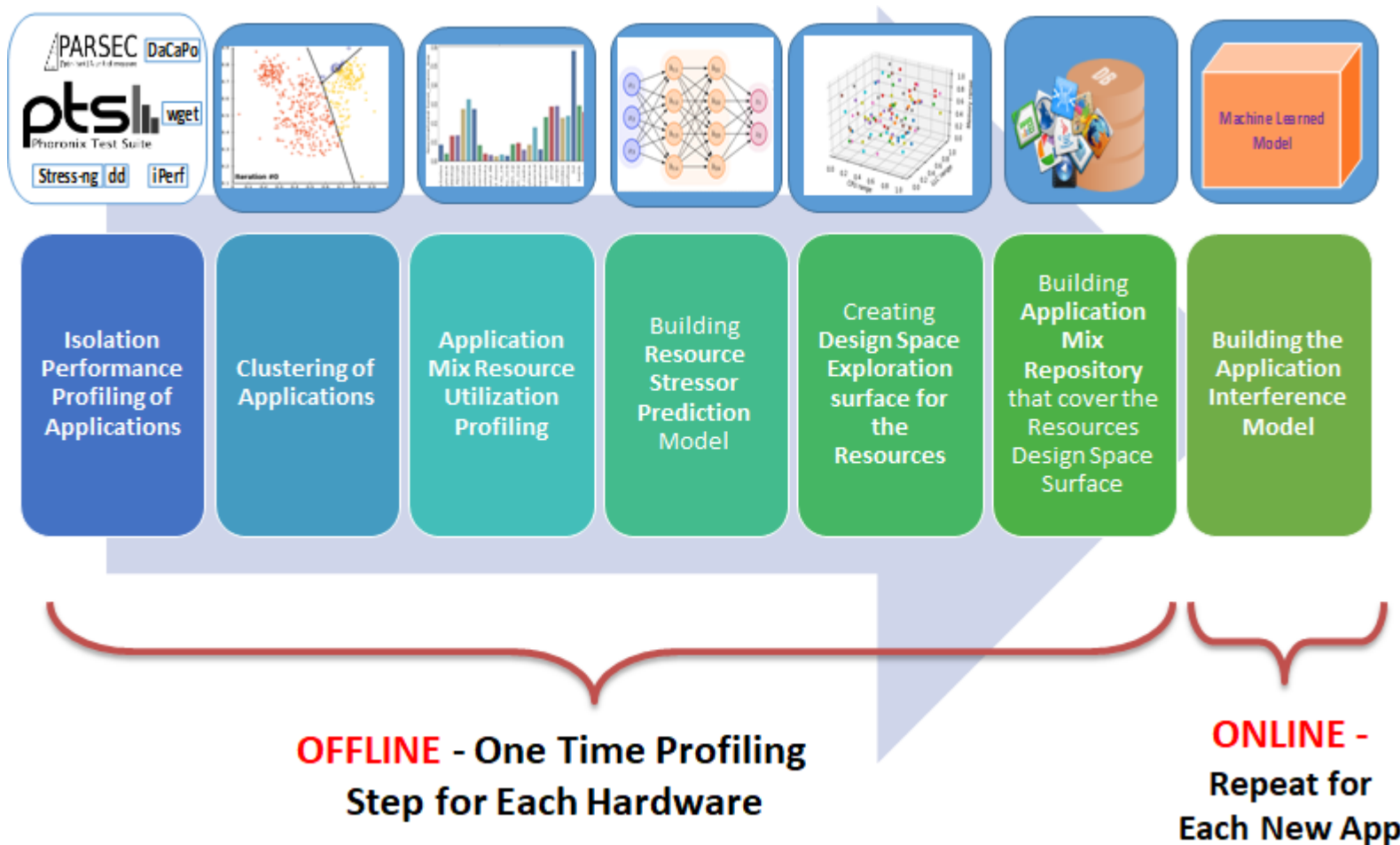
- Large number of resource dimensions and combinations
- Exhaustive profiling involves expensive monetary and time costs





# Interference-aware Performance Modeling

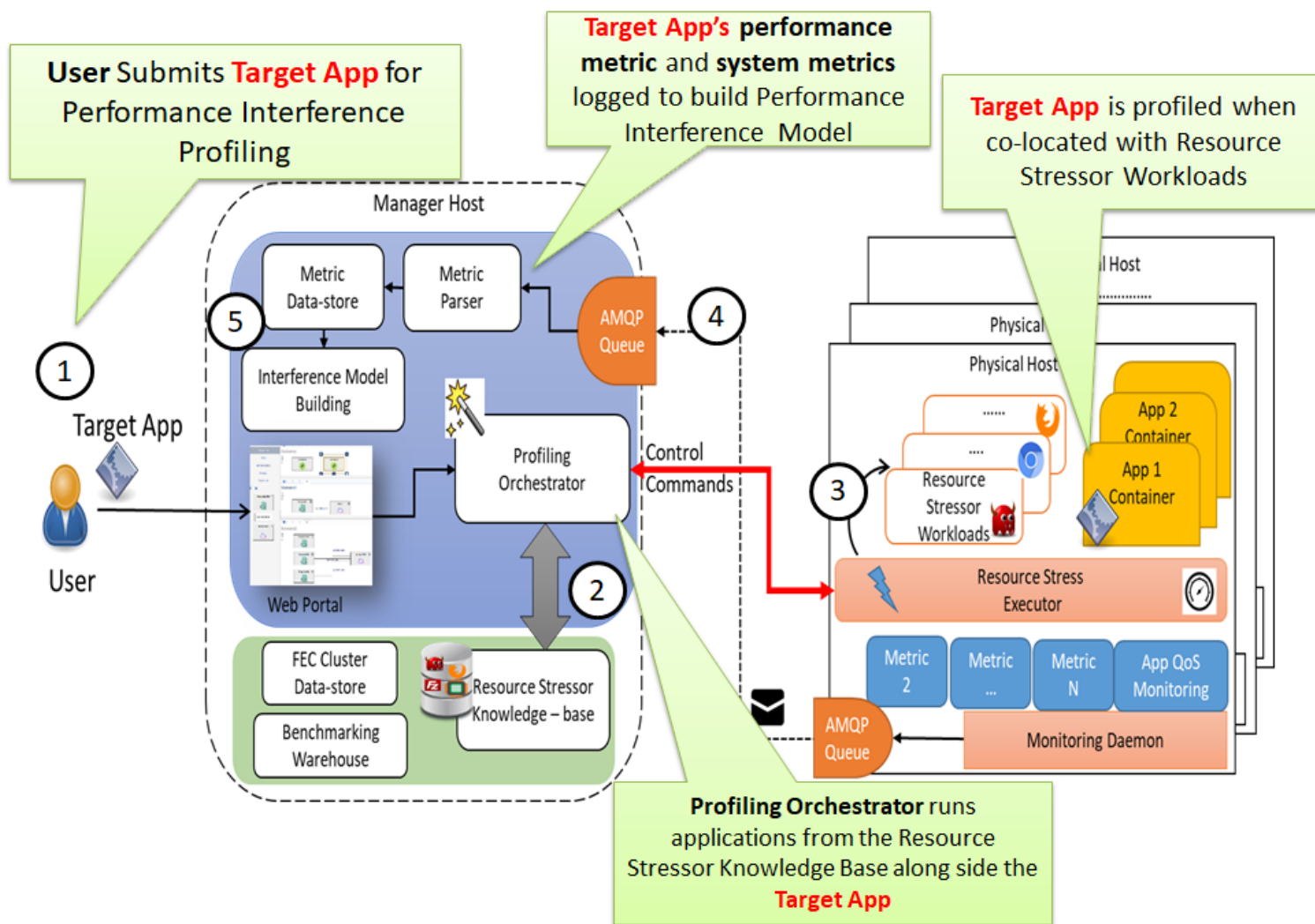
## Workflow





# Interference-aware Performance Modeling

## Online Mode





# Resource Management for Emerging Workloads





## Challenges

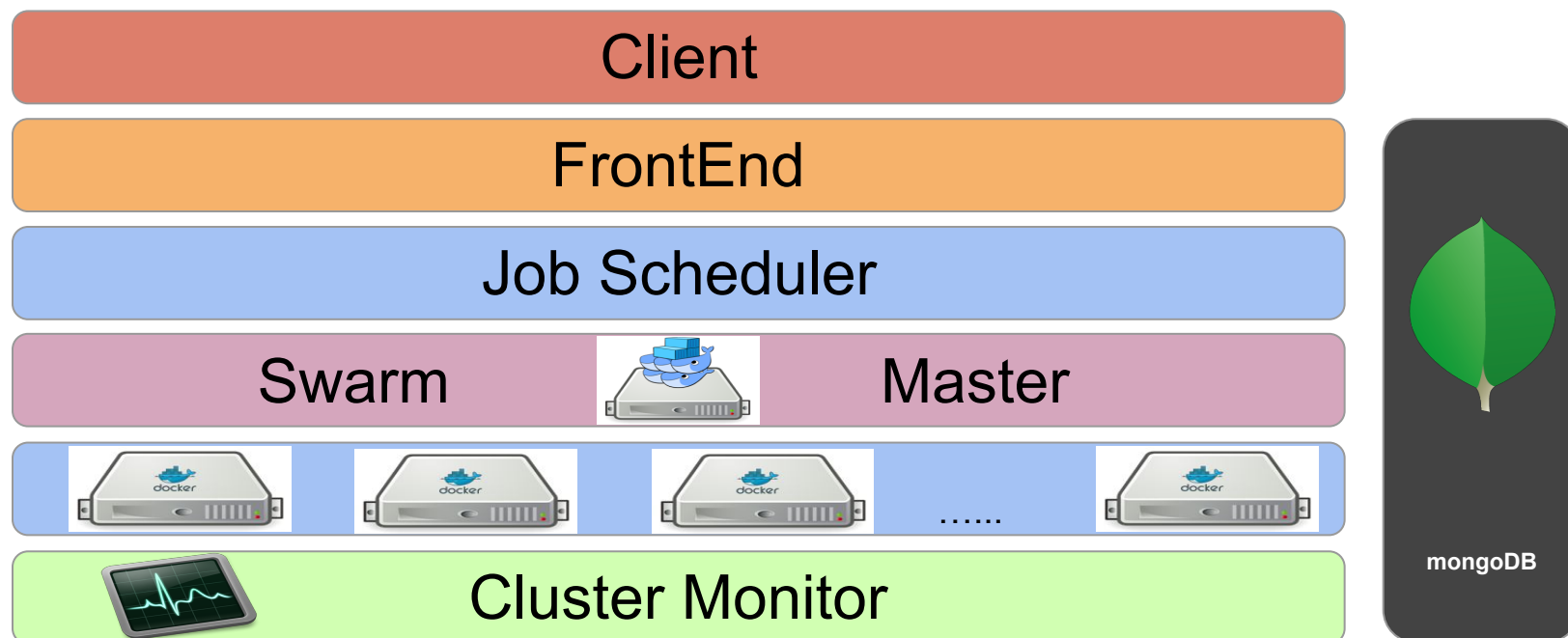
- Iterative computations
- The bulk synchronous parallel behavior
- Group scheduling or bag-of-task scheduling
- Overlay networking
- Stragglers mitigation
- Resource management schemes
- Checkpointing
- Migration of workloads





# Resource Management for Emerging Workloads

## RESTfulSwarm Solution



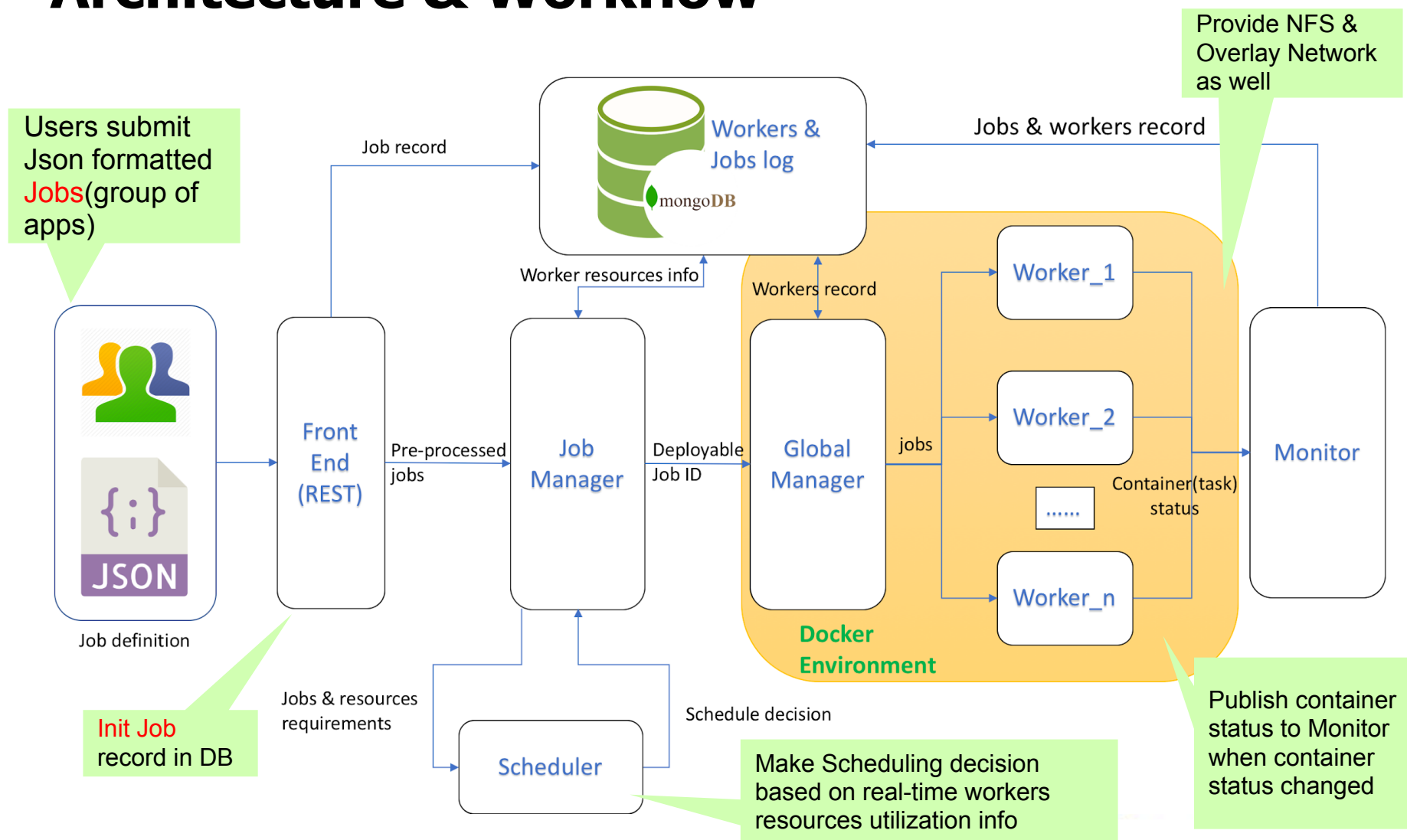
Develop a cloud platform builds on top of Docker Swarm that currently supports group scheduling schemes.





# Resource Management for Emerging Workloads

## Architecture & Workflow





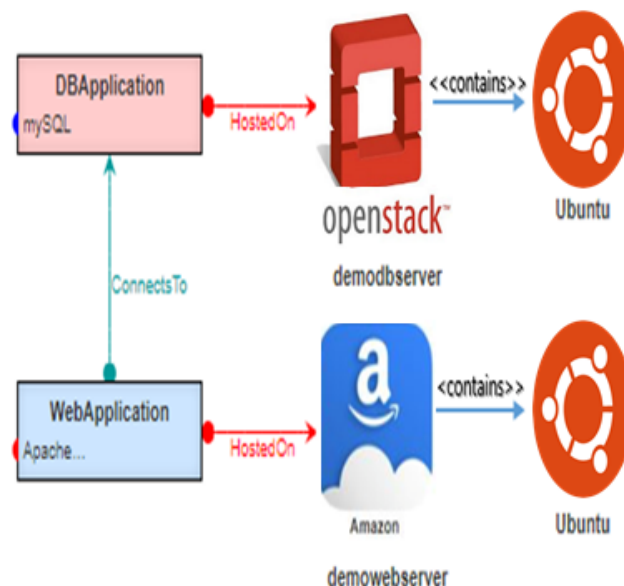
# DevOps Cloud Automation & Orchestration

## CloudCAMP Motivation

- ❖ Composite Services need be deployed on heterogeneous cloud platform.
- ❖ Users required to write Infrastructure-As-Code (IAC)
- ❖ Lack of uniform API across cloud providers

Deployment And Provisioning Involves:

1. Runtime Platform: Virtual Machine, Docker, Bare Metal
2. Software Platform: Software Package Dependencies, Software Services, Start-up Dependencies



e.g., Figure 1 shows a service topology consisting of two connected software stacks, i.e., Web front-end and MySQL database

**Objective:** GUI-based cloud automation and orchestration framework.





## Requirements

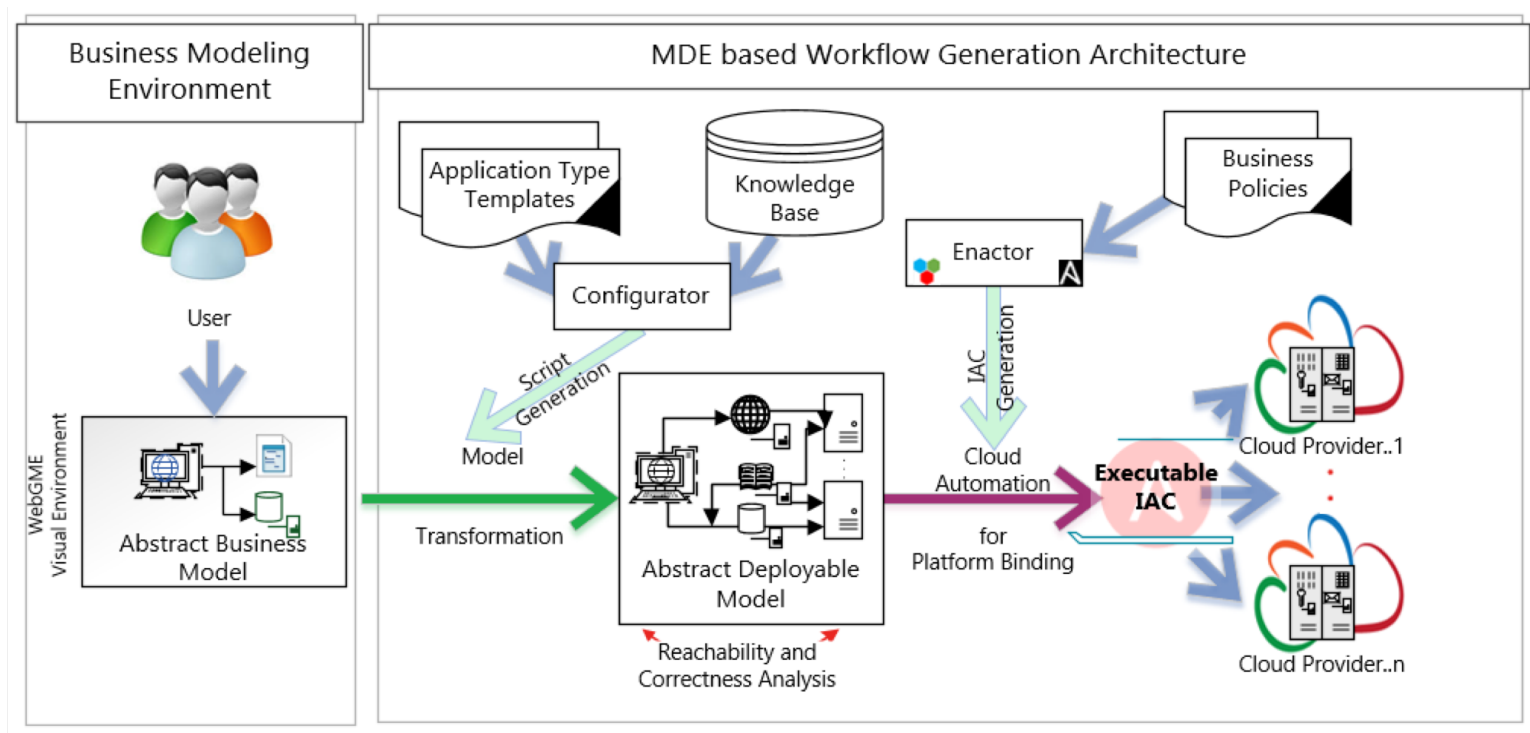
- Reduction in specification details of Cloud Infrastructure and Application Components
  - Identifying the commonalities of the cloud provisioning stacks.
  - The minimal number of variability points are the user inputs e.g. VM name, service name, VM flavor etc.
- Auto-completion for Infrastructure Provisioning using model transformation
- Support for Continuous Delivery, and Migration of service components





# DevOps Cloud Automation & Orchestration

## Architecture & Basic Workflow





FECBench: Bare Metal

RESTfulSwarm: Bare Metal

CloudCAMP: Bare Metal & KVM

In future: SDN, GPUs





# Chameleon Pros & Cons

## Pros:

- Active community
- Heterogeneous hardware
- Direct access to the bare-metal

## Cons:

- Unpredictable performance
  - VM response time is slow sometimes
  - remote access through ssh takes too long time at times
- No API access feature
  - such as AWS access key & secret access key





# Thank you!

Contact: [zhuangwei.kang@vanderbilt.edu](mailto:zhuangwei.kang@vanderbilt.edu)

