

## Motivation and Objectives

### Motivation

- Provisioning of multi-cloud resources for scientific workflows
- Loosely coupled integration with cloud management platforms
- Leverage cloud elasticity for autoscaling of scientific workflows driven by workflow execution stage

### Objectives

- Integrate the HyperFlow workflow runtime environment with the PaaSAGE cloud platform
- Application-agnostic interplay of application-specific workflow scheduler with generic provisioning and autoscaling components of PaaSAGE

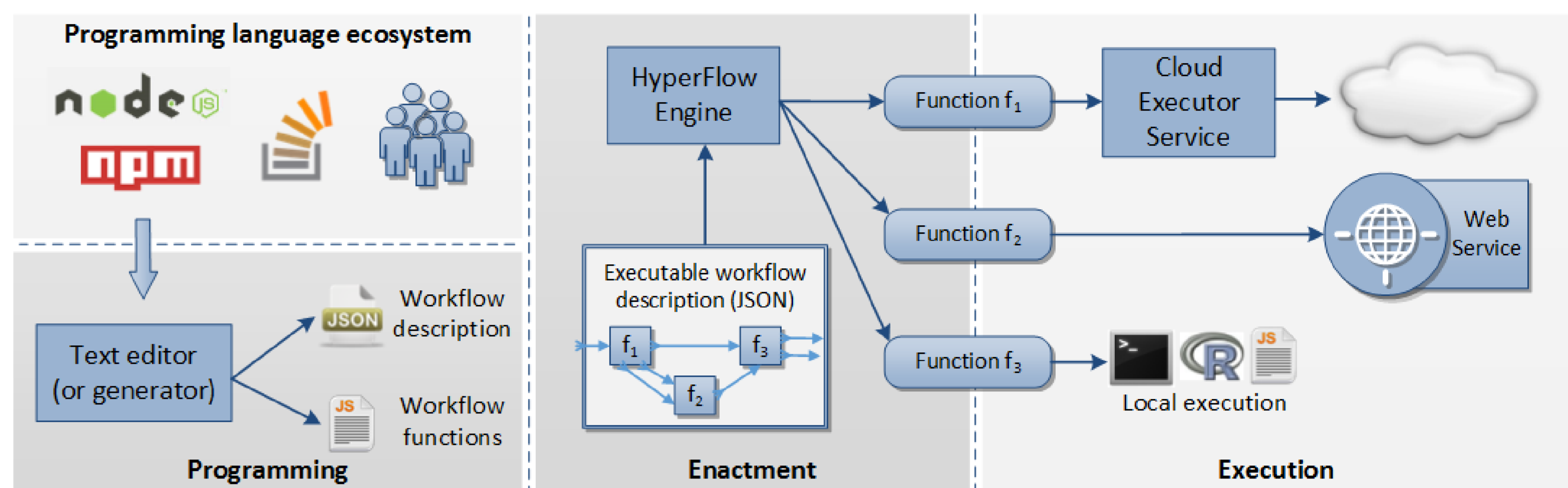
## PaaSAGE platform

Open and integrated platform to support model-driven development, deployment and adaptive execution of multi-cloud applications.

Define your application once — deploy it at the full spectrum of the Clouds

## HyperFlow workflow engine

- Simple high-level workflow description + low-level programming capabilities for advanced developers
- Skilled programmers can be as productive as in any mainstream programming language
- Lightweight, non-invasive workflow deployment model that can be applied to various cloud platforms / infrastructures



- Lightweight enactment engine based on platform Node.js, easy to deploy in the cloud, workflow engine is treated as part of the cloud application
- Abstracts details of the execution environment enabling loosely-coupled integration with execution, scheduling and resource provisioning services

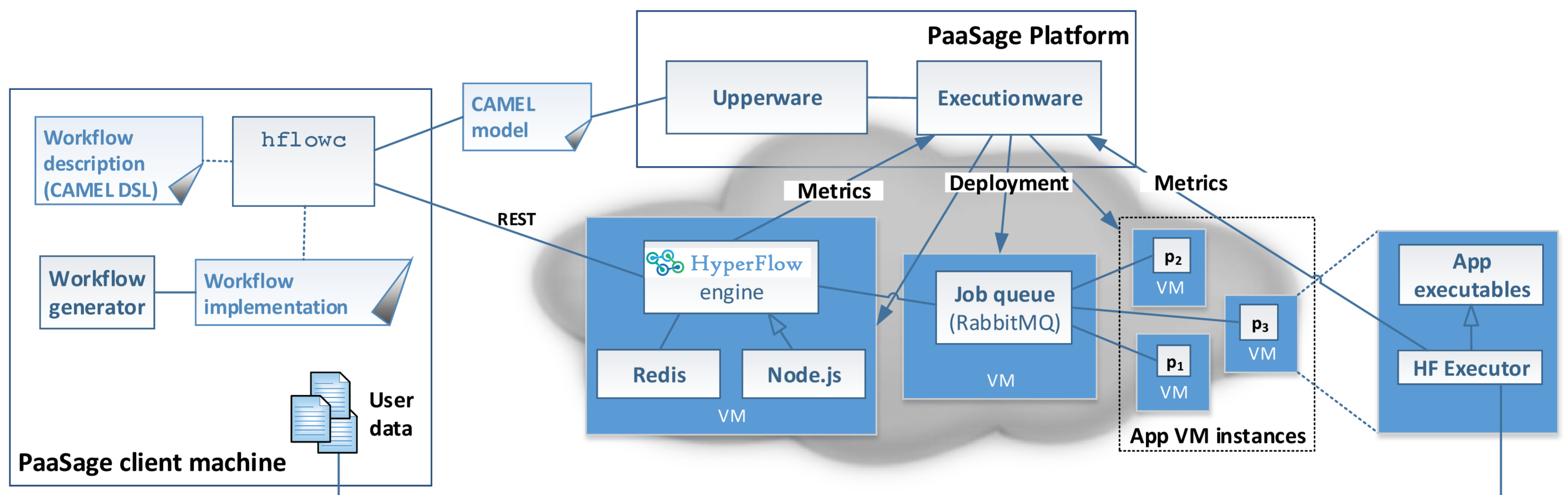
## Solution Architecture

### Main components

- **HyperFlow workflow engine**: workflow enactment
- **Job queue**: submission of task requests, fetching results, and execution monitoring
- **hflowc**: prepares the execution plan taking into account constraints such as cost and deadline and generates the CAMEL model
- **Executors**: deployed on VMs alongside workflow components, controlling their execution and data transfer

### Rapid deployment model and its benefits

- Improved isolation: each workflow runs in its own sandbox (separate instance of the workflow runtime system)
- Improved performance: workflow engine orchestrates the workflow from inside the cloud leveraging instance-to-instance communication.
- Easier integration: no tight coupling integration with a specific cloud platform is required; the same deployment model can be applied to various clouds through deployment plugins



### Integration with PaaSAGE

- **CAMEL application model** automatically generated based on the HyperFlow workflow description. Includes initial deployment plan and scalability rules which control autoscaling behavior
- **Monitoring information** sent from the Task scheduler and VM workers to the PaaSAGE Executionware; Triggers the scalability rules and automatic scaling of the workflow application

### Conclusion

- A solution for deployment, execution and autoscaling of scientific workflows
- Workflow runtime environment deployed in the cloud on-demand as part of the workflow application
- Loosely-coupled integration with the PaaSAGE cloud platform

## References

1. B. Baliś, "HyperFlow: A model of computation, programming approach and enactment engine for complex distributed workflows". Future Generation Comp. Syst, 55:147-162, 2016
2. M. Malawski, K. Figiela, M. Bubak, E. Deelman, and J. Nabrzyski: "Scheduling multi-level deadline-constrained scientific workflows on clouds based on cost optimization", Scientific Programming 2015
3. T. Dziok, K. Figiela, M. Malawski: "Adaptive multi-level workflow scheduling with uncertain task estimates", PPAM 2015
4. B. Baliś, K. Figiela, M. Malawski, M. Pawlik, and M. Bubak. "A lightweight approach for deployment of scientific workflows in cloud infrastructures", PPAM 2015



PaaSAGE: Model-Based Cloud Platform Upperware –

We thankfully acknowledge the support of the European ICT-FP7 program through the PAASAGE (IP 317715) project