

Cloud Foundry Architecture

Team: "Pipes_and_Filters" {

Brett_Borchardt

Marc_Johnson

Paul_Kleczka

Allan_Tokuda

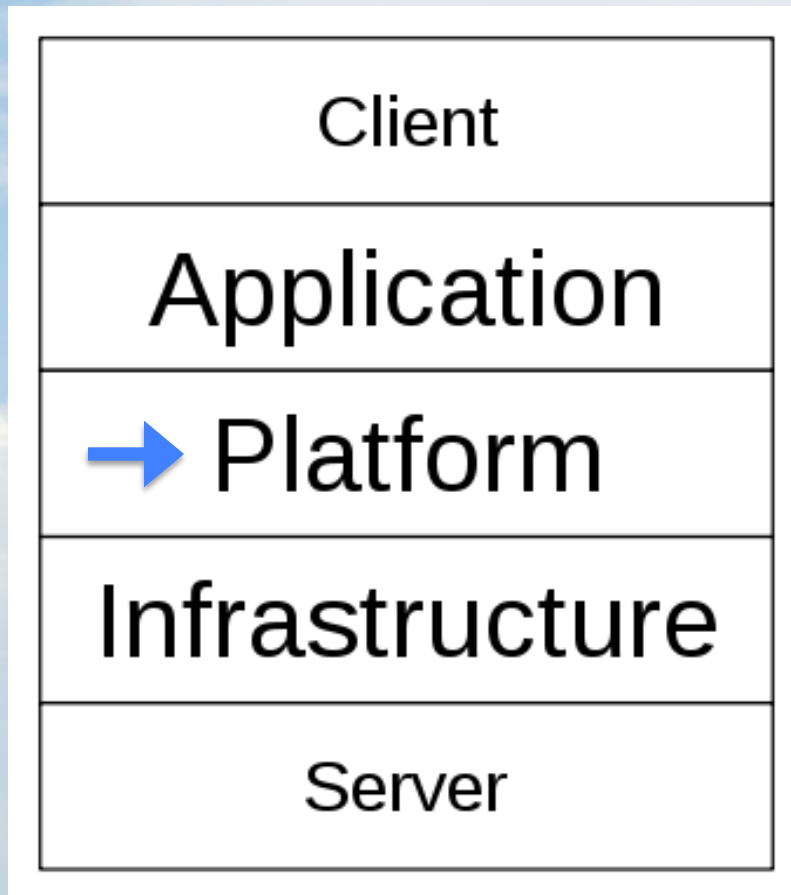
}



Outline

- Intro
- Goals and Principles
- Functional View and Scenarios
- Concurrency View
- Perspectives
- Proposed Extension

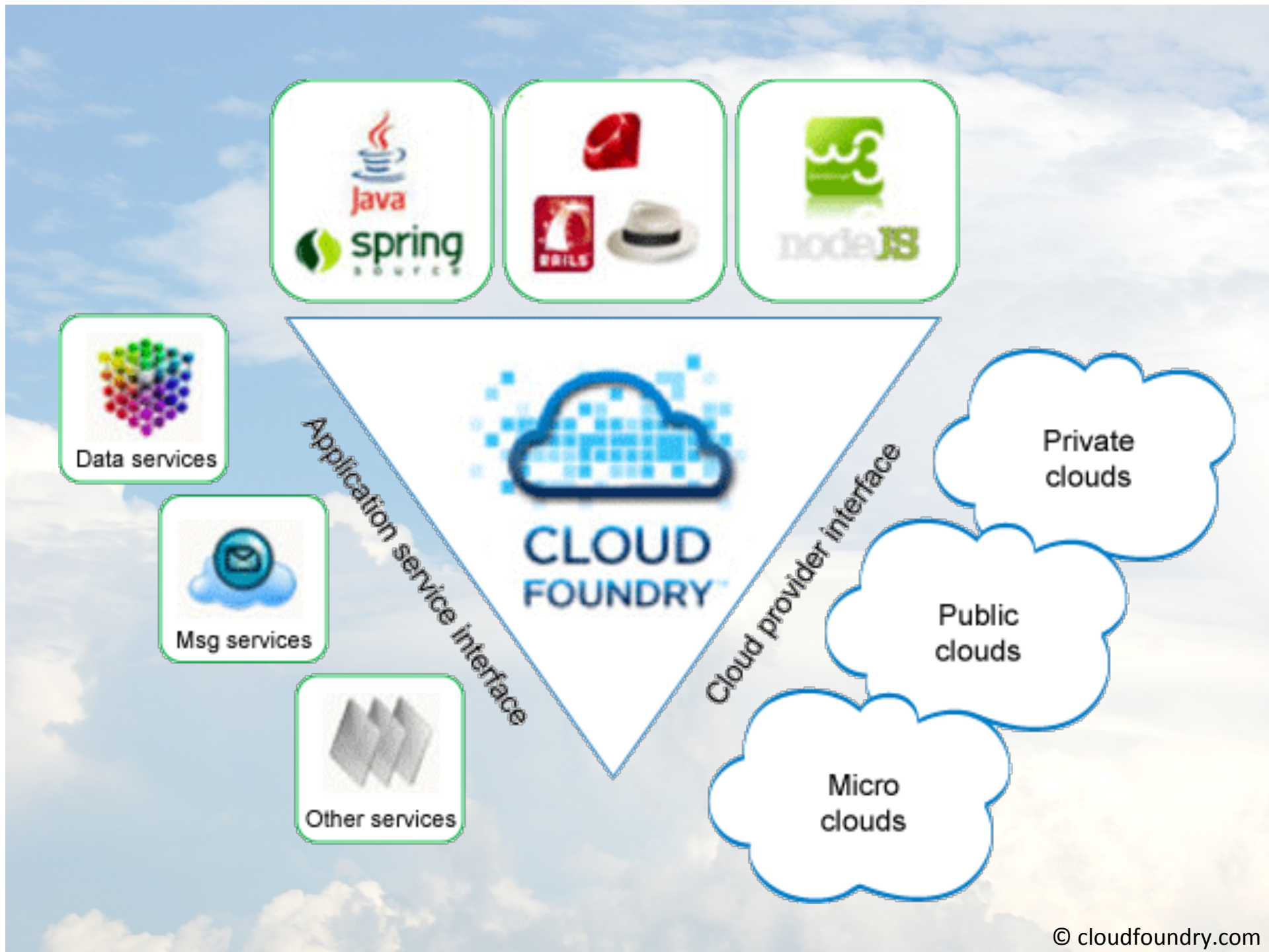
What is Cloud Foundry?



SAAS (Custom App)

PAAS (Cloud Foundry)

IAAS (AWS)



Quality Perspectives

- Performance and Scalability
 - Horizontal scaling of apps/services
 - Ruby fibers
- Availability and Resilience
 - Multi-tenant isolation
 - Health manager
- Evolution
 - Loosely coupled interfaces via messaging/REST

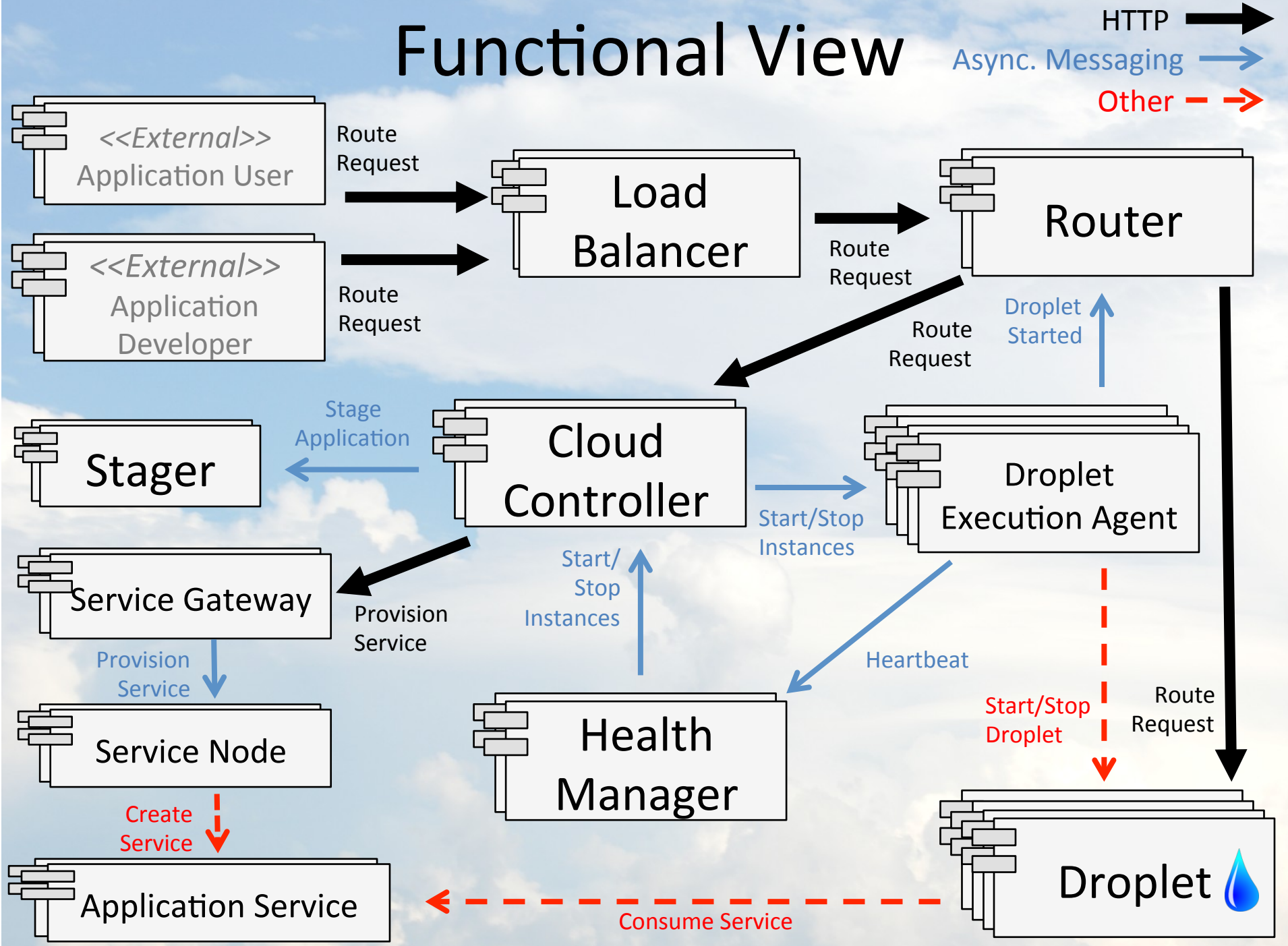
Architectural Principles

- Principle 1: Open platform
- Principle 2: Extensible architecture
- Principle 3: Positive developer experience
- Principle 4: Simple design
 - Low efferent coupling
 - RESTful communication and async. messaging
 - Idempotent service interfaces

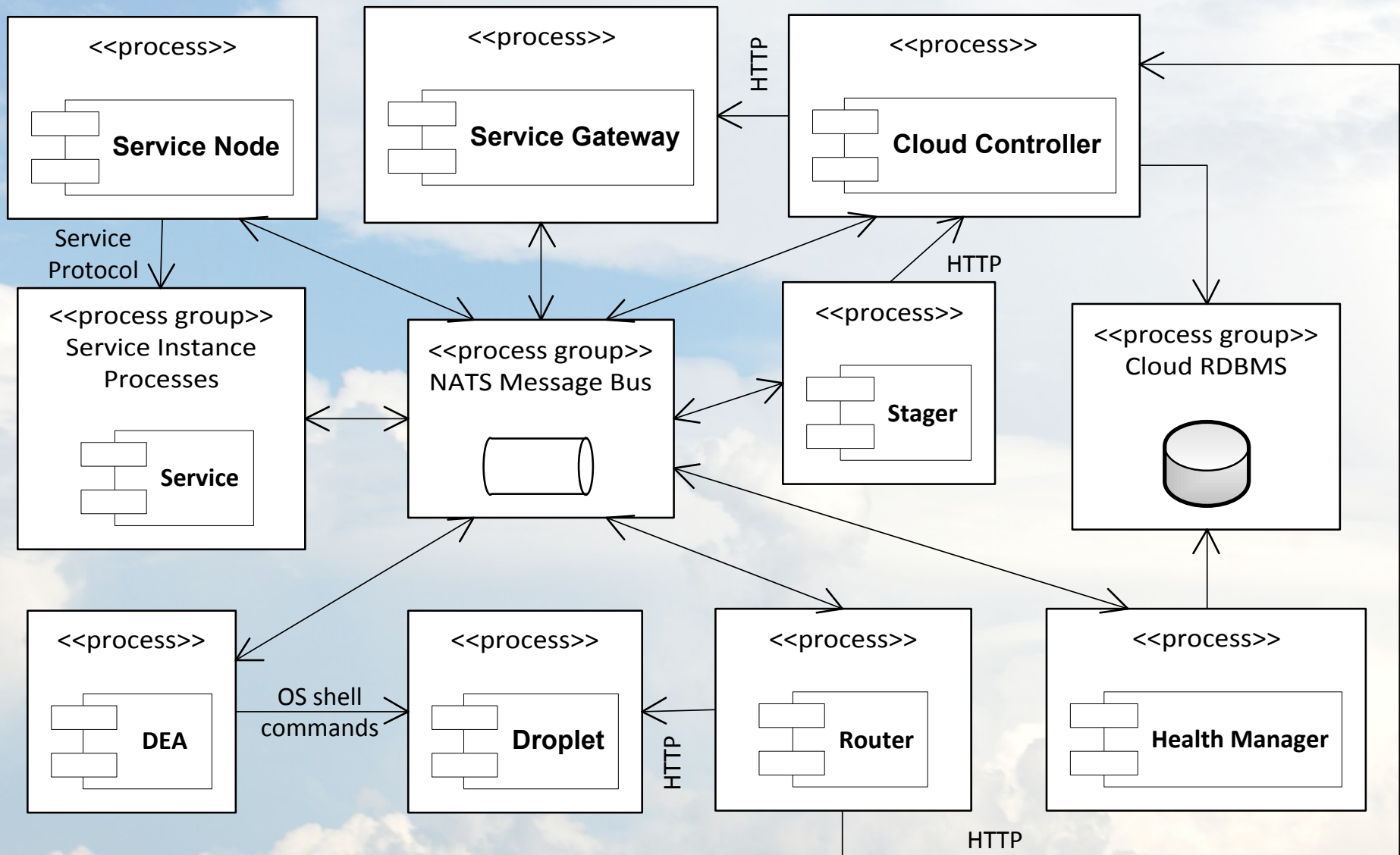
Functional Scenarios

- Target and Authenticate
 - Deploy Application
 - Provision and Bind Service
 - Start/Stop Application
- } Application Developer
- Application Request by End User

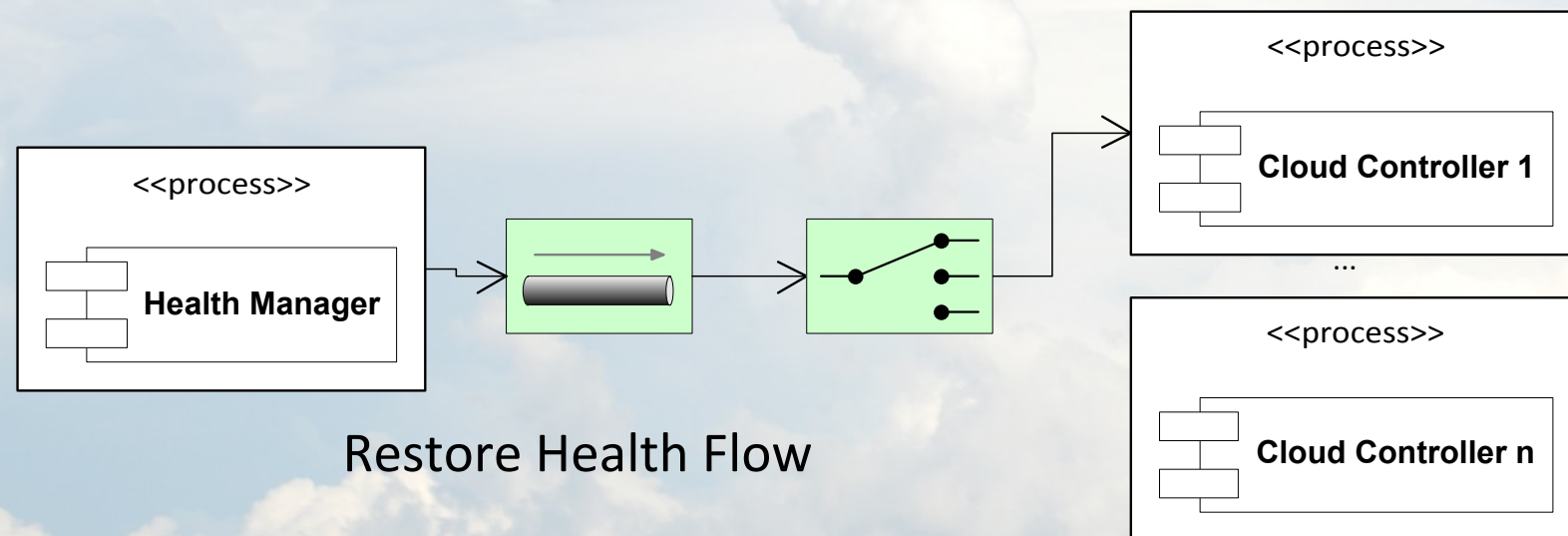
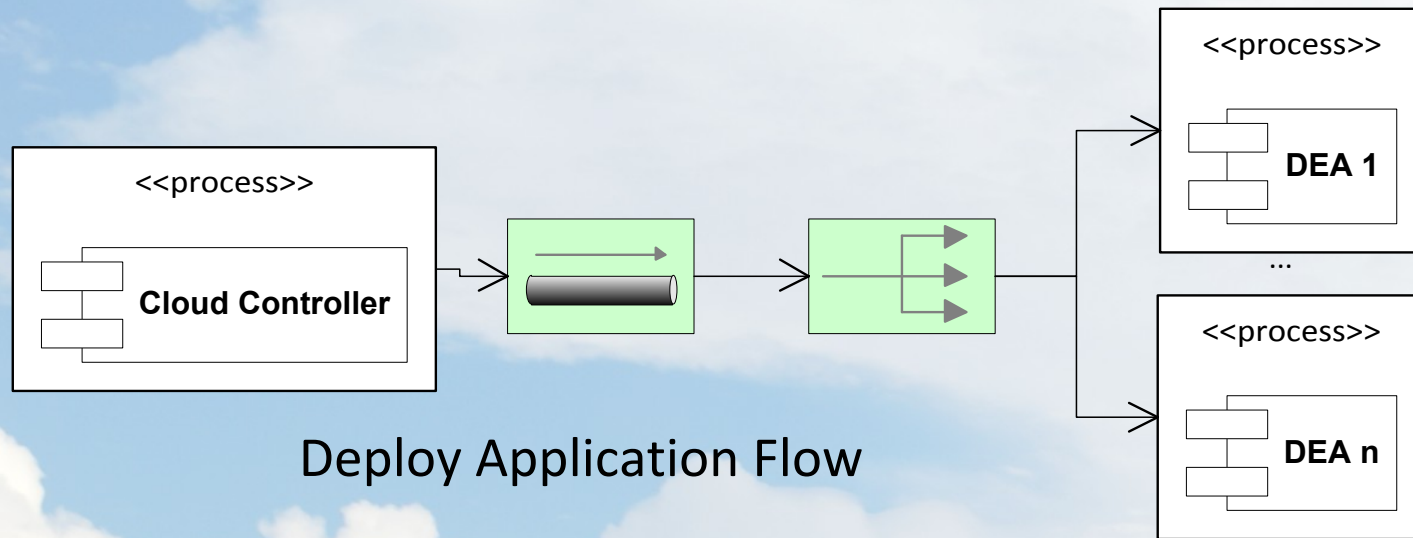
Functional View



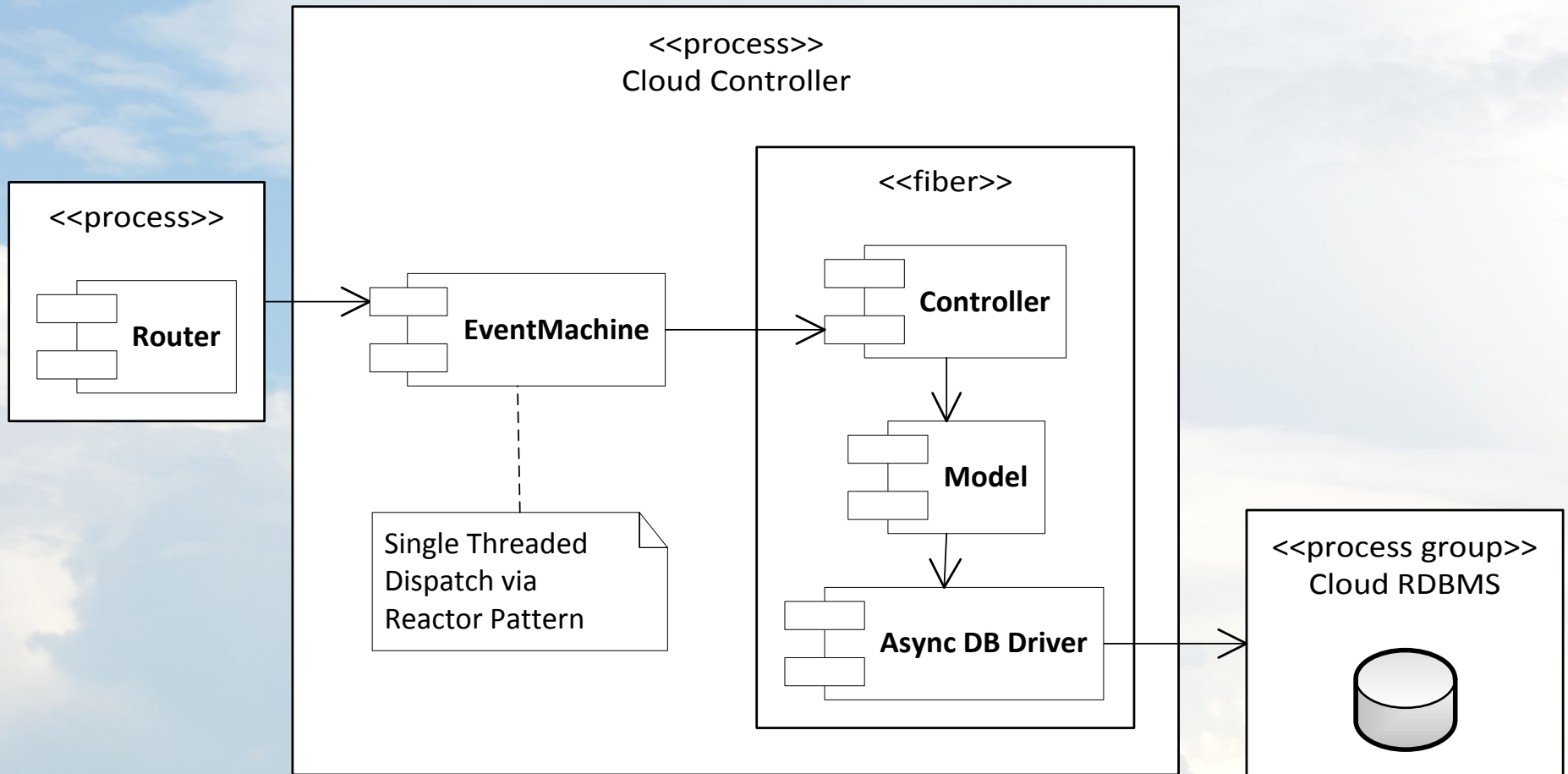
Concurrency: Communication & State



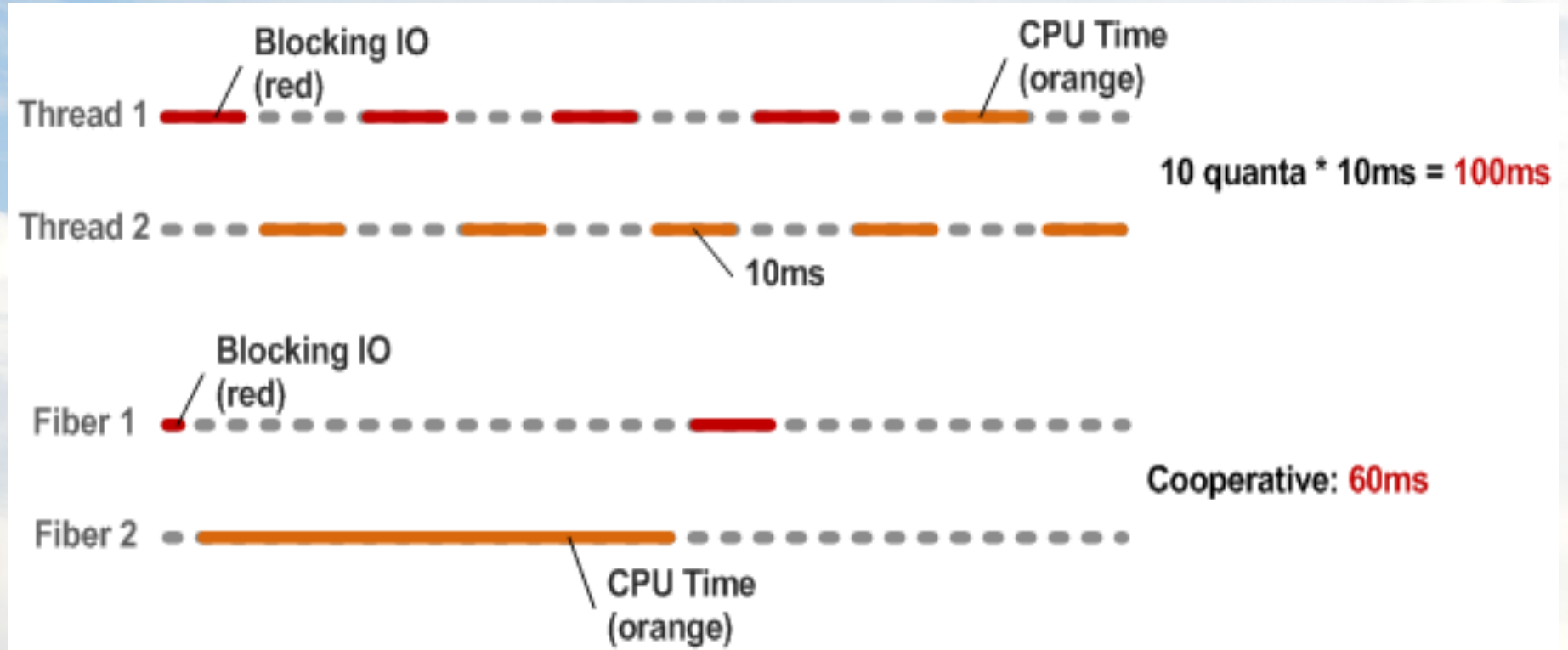
Concurrency: Messaging



Concurrency: Fibers



Concurrency: Fiber Scheduling



Source: <http://www.igvita.com/2009/05/13/fibers-cooperative-scheduling-in-ruby/>

Proposed Extension

- **Auto-scaling:** provide detailed means of controlling how many instances of an app are running
 - Schedule regular increases and decreases
 - Respond to demand pickup/dropoff
 - Decide from system metrics
 - Propagate scaling decisions





Questions?

Brett Borchardt

bborchardt@gmail.com

Marc Johnson

marc.e.johnson@gmail.com

Paul Kleczka

kleczka@gmail.com

Allan Tokuda

allan.tokuda@gmail.com

References

- [1] Cade Metz, **Man Survives Steve Ballmer's Flying Chair To Build '21st Century Linux'**,
<http://www.wired.com/wiredenterprise/2011/11/cloud-foundry/all/1>
- [2] [3] Derek Collison - **Distributed Design and Architecture of Cloud Foundry**
<http://www.slideshare.net/derecollison/design-of-cloud-foundry>

Backup Slides



Goals

- Developer Productivity
 - wide variety of frameworks and services
 - seamless integration into applications
 - simple deployment model and toolset
- Open System
 - Extensible frameworks + services.
 - Runs public, private, or hybrid
 - Runs on a virtual machine
- Faster Delivery
 - Simple and high performance framework
 - Third-party (or enterprise) integration
 - “Micro” cloud on developer machine

What is Cloud Foundry? (Intro)

- **Multi-Language**

- Ruby, Java, Scala, Node.js, Erlang, Python, PHP..

- **Multi-Framework**

- Rails, Sinatra, Spring, Grails, Express, Lift

- **Multi-Services**

- MySQL, Postgres, MongoDB, Redis, RabbitMQ

- **Multi-Cloud, Multi-IaaS**

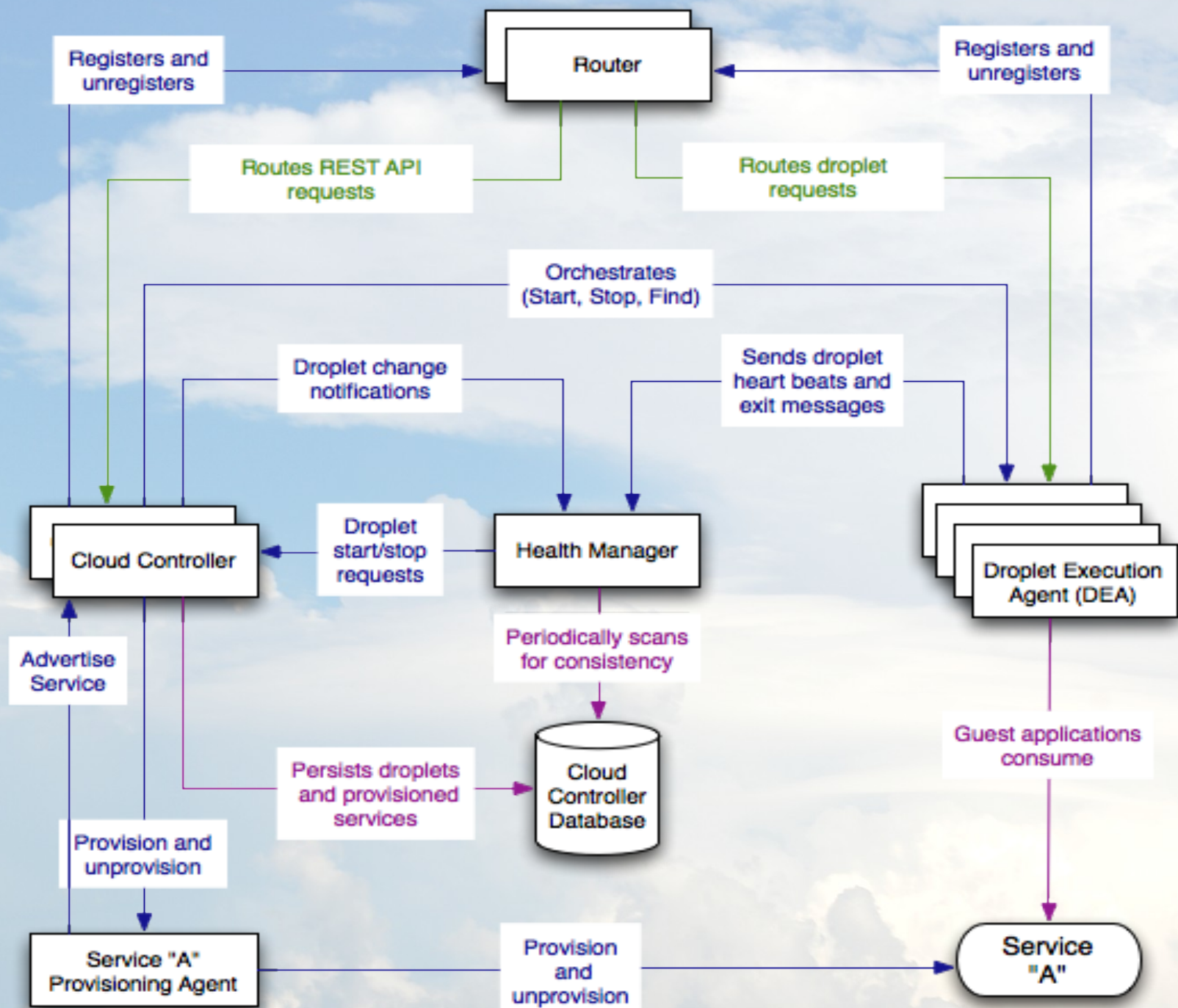
- vSphere, MicroCloud, OpenStack, AWS

What is Cloud Foundry? (Intro)

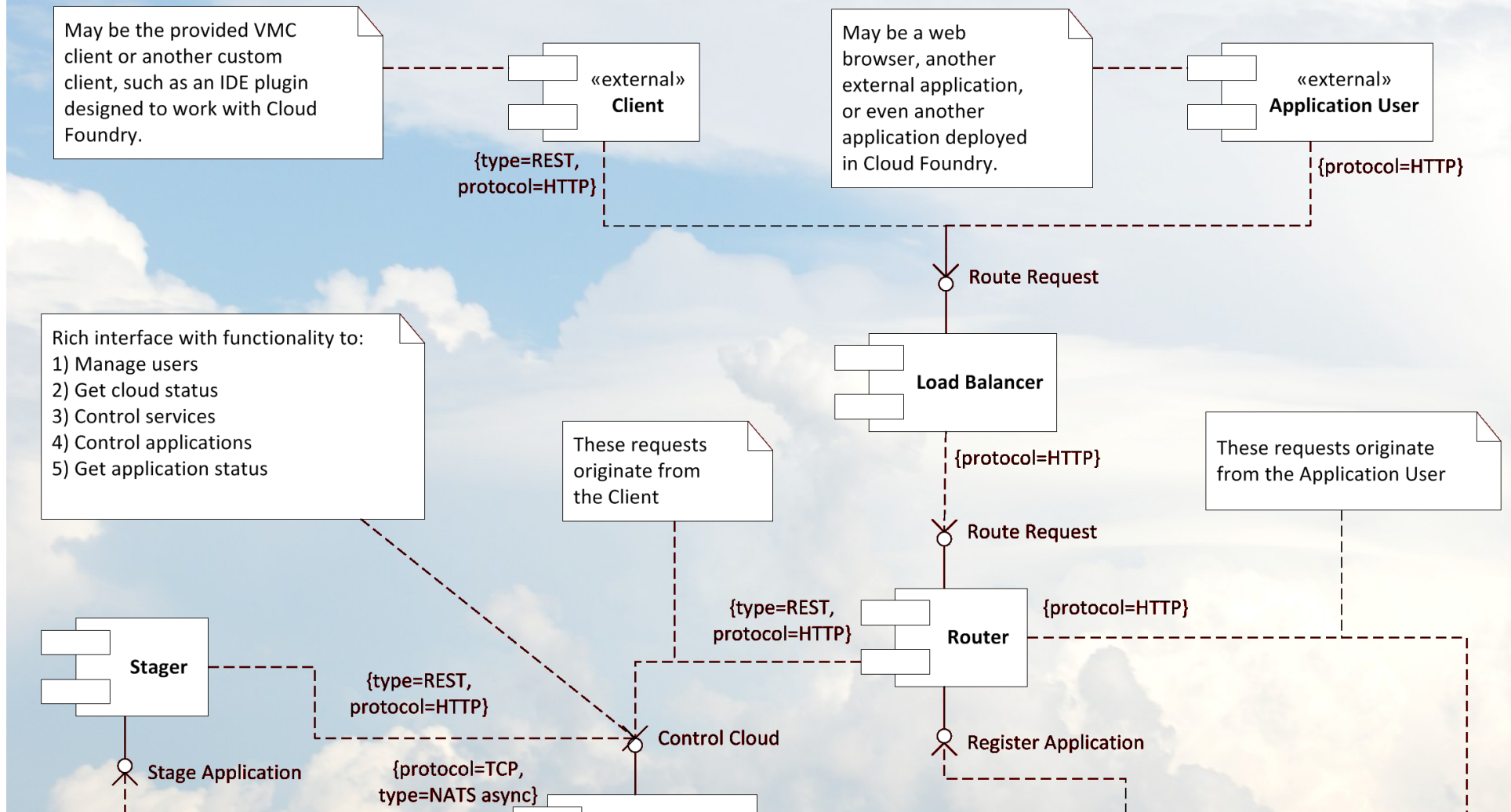
- **Open Platform as a Service (PAAS)**
- **“Linux of the Cloud” :**
 - Makes deploying and scaling fast and easy
 - Open source (written in Ruby)
 - Supports multiple development frameworks , extensible
 - Public clouds, private clouds, and “micro-cloud” which can be run on a PC

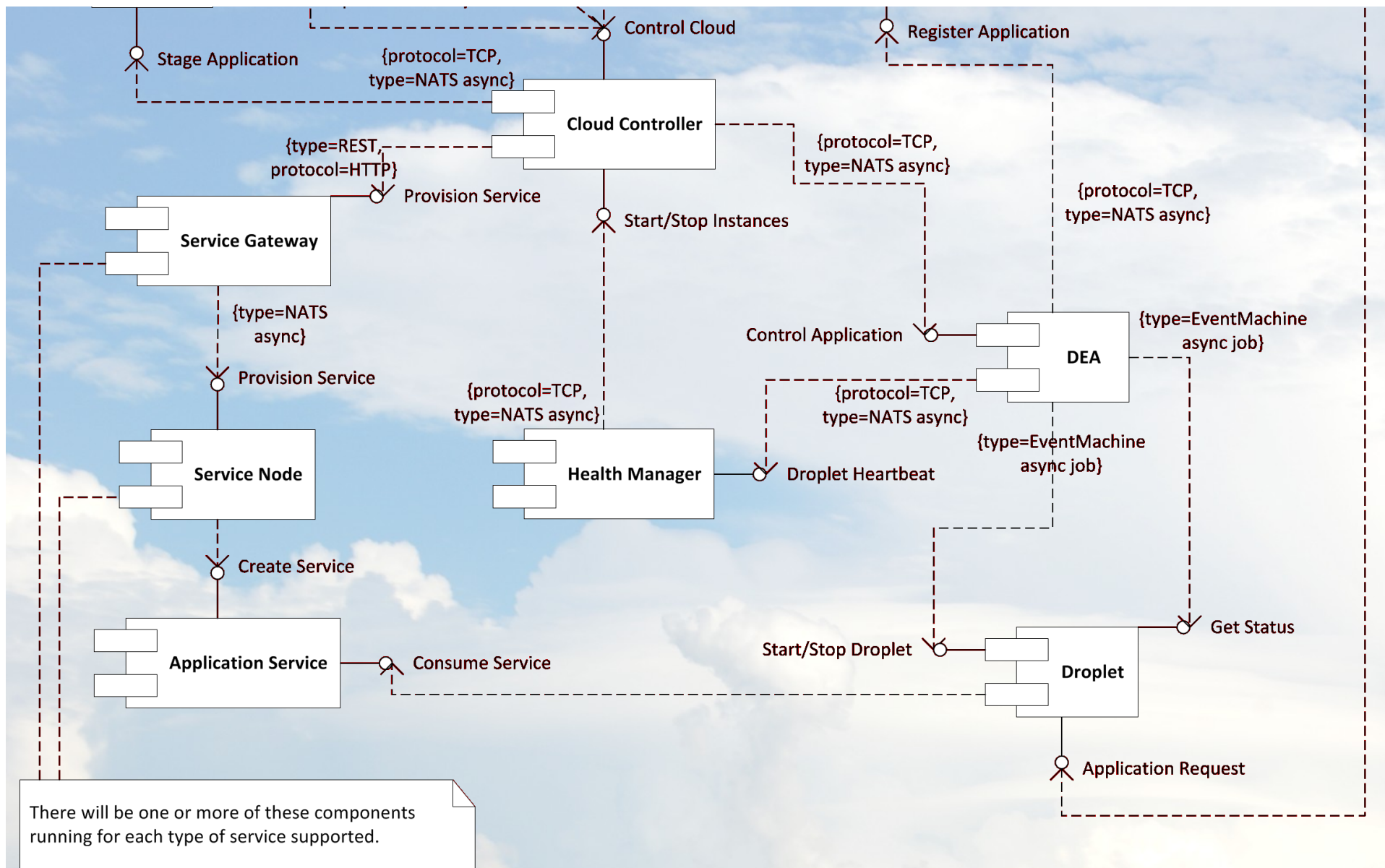
Not VMs, Memory, Storage, Networks, CPU

Functional View



Functional View





Functional View

