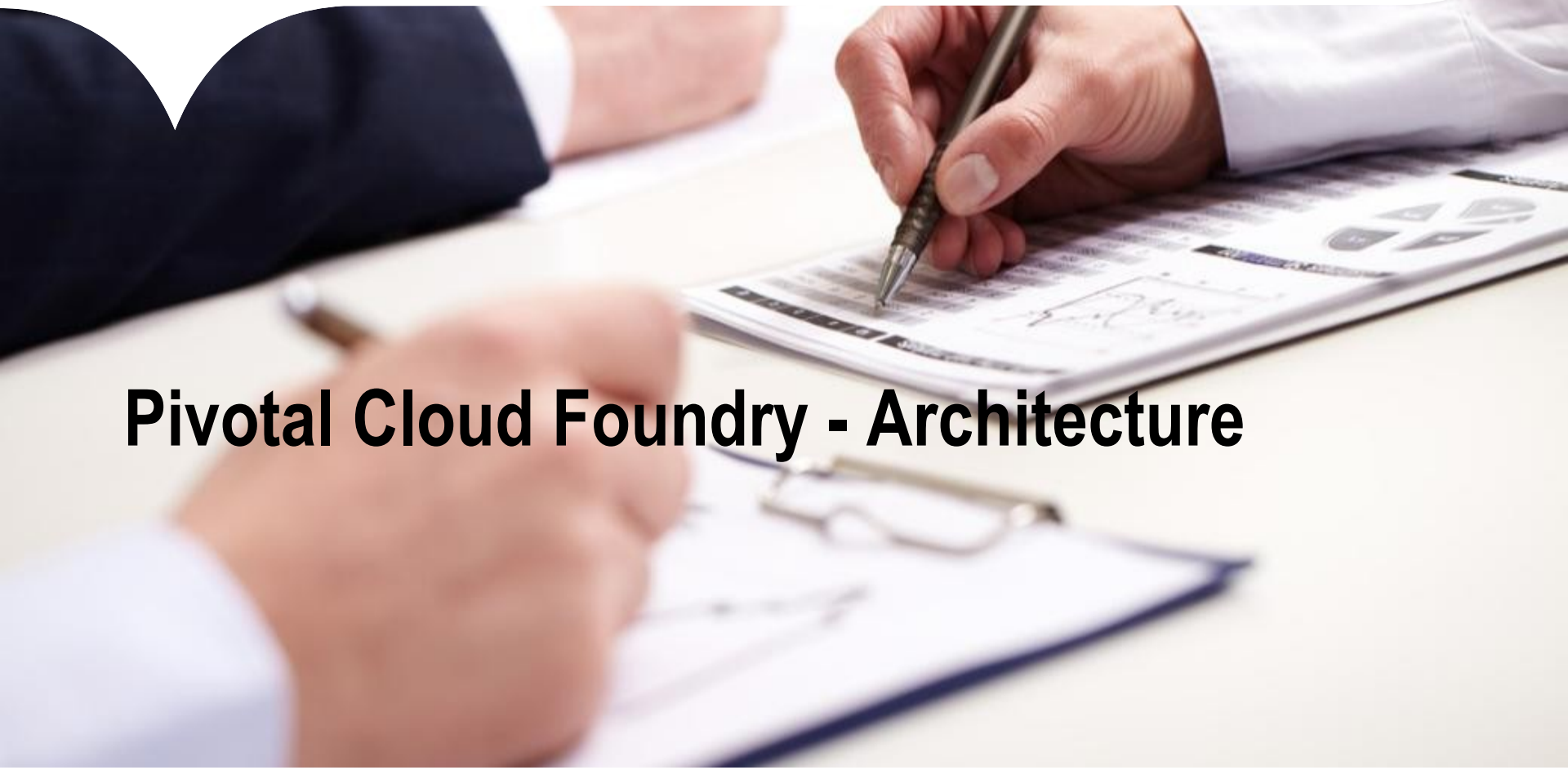
Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING



Pivotal Cloud Foundry - Architecture

Objectives of PCF

- Purpose:
  - To learn pivotal cloud foundry architecture.
- Product:
  - Components
  - Health Manager
- Process:
  - To understand the Cloud Foundry Architecture

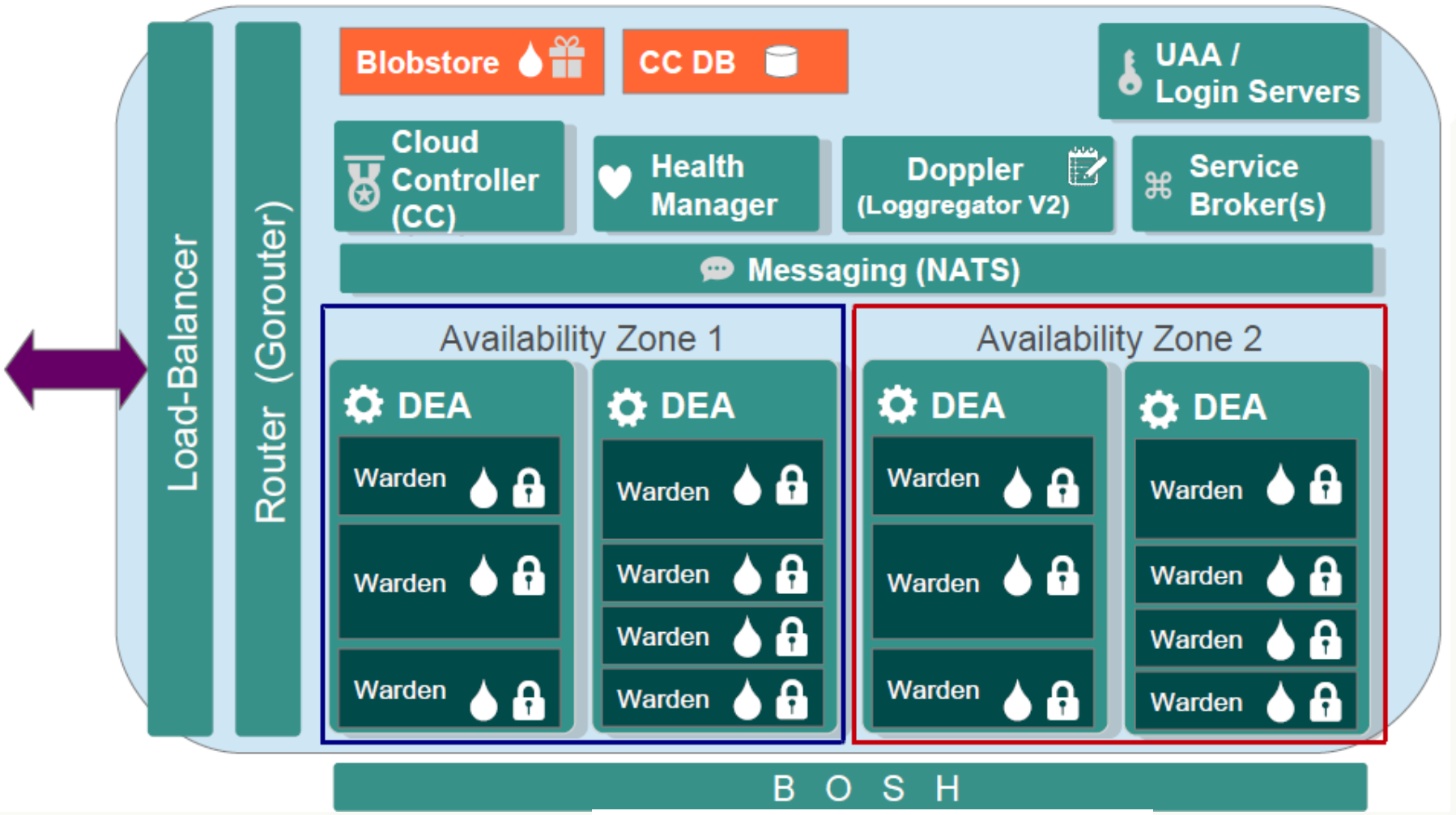
Pivotal Cloud Foundry Architecture

- Pivotal CF Components
- Load Balancer (HA Proxy)
- Router
- Cloud Controller
- BlobStore
- Cloud Controller Data Base(CC DB)
- Avalilability Zones

- Droplet Execution Agents(DEA)
- Warden Container
- Service Broker
- Health Manager
- Messaging (NATS)
- Loggregator
- UAA and Login Services
- Cloud Foundry BOSH

Pivotal Cloud Foundry Architecture

Pivotal CF Components – 1.4



DEA → Droplet Execution Agent
UAA → User Authorization & Authentication

Load Balancer (HA Proxy)

Description

- HAProxy is separate component from the Router
  - Sits in front of the Router
  - Only one instance allowed
    - Single – point of failure
- Cloud Foundry can only have one IP address visible to outside
  - This is the Load Balancer
  - It passes requests to the Router(s) to send the right CF component

Responsible For

- Load – Balancing across multiple routers (if used)
- SSL termination
- **Note:** Typically need to configure your own HAProxy or other load-balancer (such as F5 or NSX) for improved performance SSL handling and high availability.

Router

Description

- Routes all incoming HTTP traffic
  - System traffic (cf commands)
  - Application traffic
- Maintains dynamic routing table for each load-balanced app instance
  - Knows IP addresses and ports.
- Multiple routers possible
  - Configurable by admin in Ops Manger
- From CF 1.4 rewritten in Go – the *GoRouter*

Responsible For

- Load balancing across application instances
- Maintaining an active routing table
- Access logs
- Supports web-sockets

- In CF 1.6 will also do SSL termination

Cloud Controller

Description

- Command and Control
  - Responds to clients (CLI, web UI, Spring STS)
  - Account and provisioning control
- Provides RESTful interface to domain objects
  - Apps, services, organizations, spaces, service instances, user role , and more …
- Multiple Cloud Controllers possible
  - Configurable by admin

Responsible For

- Expected Application state, state transitions, and desired convergence
- Permissions/ Authorization
- Organizations / Spaces / Users
- Service management
- Application placement
- Auditing / Journaling and billing events
- Blob storage

BlobStore

Description

- Storage for binary large objects
- Eliminates need for upload / re-staging when scaling applications
- Currently NFS mounted storage
  - Or Amazon S3 store

Responsible For

- Store uploaded application packages (cf push)
- Stores Droplets

Cloud Controller DataBase

Description

- Storage for application metadata
- Used exclusively by the Cloud Controller

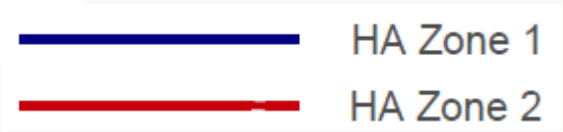
Responsible For

- Stores information on
  - Application name
  - # of instances requested
  - memory limit
  - Routes
  - Bound Services
- A Postgres DB instance

Availability Zones

Description

- Separate shared points of failure such as a power, network, cooling, roof, floor.
- Allows Cloud Controller to locate instances on separate zones to boost redundancy.
- Marked on diagram as:



Responsible For

- Enhancing application redundancy
- Provides one of the layers of High Availability

Droplet Execution Agents (DEA)

Description

- Secure and fully isolated containers
  - Actually a Linux VM
- Responsible for an apps lifecycle
  - Building, starting and stopping Apps as instructed
- Periodically broadcast messages about their state
  - Via the NATS message bus.
- Typically many DEAs in a Cloud Foundry installation

Responsible For

- Managing Linux containers
  - Pivotal's Warden containers
- Monitoring resource pools
  - Process
  - File System
  - Network
  - Memory
- Managing app life cycle
- App log and file streaming
- DEA heartbeats
  - Via NATS to Cloud Controller & Health Manager

Warden Container

Description

- Isolated Process
  - Safe, lightweight alternative to full VM
  - Runs a Droplet
- Pivotal's secure implementation of LXC (Secure Linux Container)
- Isolates application from each other
- Allows multiple applications running on each VM

Responsible For

- Isolates application running on the same VM
  - Individual failures does not affect other applications on the VM
  - Uses kernel namespaces to isolate network, disk, memory and CPU
  - Uses Linux *cgroups* to do resource management
- Secures applications from environment
- Runs Droplets

Service Broker

Description

- Provide an interface for native and external 3rd party services
  - Service processes run on Service nodes
  - Or with external as-a-service providers
- Examples
  - Mail server
  - Database
  - Messaging
  - Many more …
- Typically one service broker for each marketplace service.

Responsible For

- Advertising service catalog
- Makes create /delete/ bind / unbind calls to service nodes
- Request inventory of existing instances and bindings from cloud controller for caching, orphan management
- SaaS marketplace gateway

Health Manager

Description

- Monitors application uptime
- Listens to NATS message bus for mismatched application states (expected vs. actual)
  - Cloud Controller publishes expected state
  - DEAs public actual state
- state mismatches are reported to the Cloud Controller.
- Multiple Health Managers possible

Responsible For

- Maintains the **actual state** of apps
- Compares to **expected state**
- Sends suggestions to make actual match expected
  - Cannot make state changes itself – only Cloud Controller can do that!

Messaging (NATS)

Description

- Fast internal messaging bus
- Manages system – wide communication
- Uses a publish-and-subscribe mechanism
- *Not – Another Transport system*
- Single NATS per Cloud Foundry installation

Responsible For

- Non- persistent messaging
- Pub/Sub
- Queues (app events)
- Directed messages (INBOX)

Loggregator

Description

- Master logging process
  - Accepts logs from application instances
  - Accepts logs from other CF components
- Make log data available to external sinks
- Uses Diego's Doppler Server since CF 1.4

Responsible For

- Non-persistent, temporary log-storage
- Accumulates logs from multiple storage and aggregates by application
- Provides logs to external sources such as cf logs or App manager console
- Supports log "drains" to syslog servers

UAA and Login Services

Description

- UAA =“ User Authorization and Authentication”
- Provides identity, security and authorization services.
- Manages third party OAuth 2.0 access credentials
- Can provide application access & identity-as-a-service for CF apps
- Composed of
  - UAA Server
  - Command Line Interface
  - Library
- Multiple UAA/Login Servers possible

Responsible For

- Token server
- ID Server (User Management)
- OAuth Scopes (Groups) and SCIM
- Login Server
  - UAA Database
  - SAML support (for SSO integration) and Active Directory support with the VMWare SSO Appliance
- Access auditing

Cloud Foundry BOSH

Description

- Tool Chain for managing large scale distributed systems
  - Release engineering
  - Deployment and lifecycle management
- Continuous and predictive updates with minimal downtime
- Control primitive (CPI) written for each underlying infrastructure provider
  - The ‘glue’ between CF and underlying IaaS

Responsible For

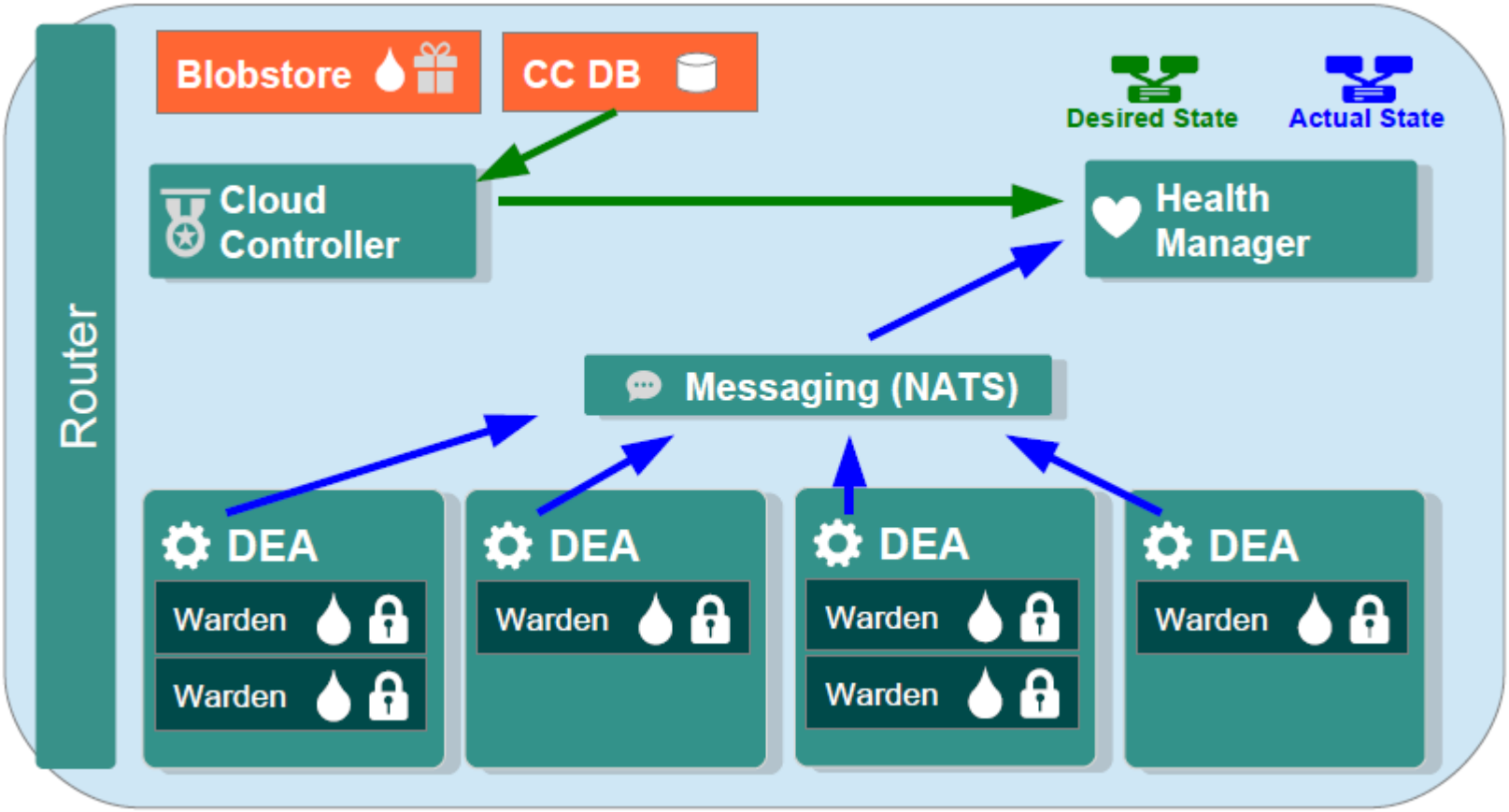
- IaaS installer
  - Currently vSphere
- VM creation and management
- Continuous and predictive updates with minimal downtime
- High Availability
  - Restarts failed CF *internal* processes, DEAs
- CPI (Cloud Provider Interface) to control underling infrastructure (IaaS) primitives

Health manager

- Monitor applications to determine version, current state and number of instances
  - Actual state of an application is based on heartbeats from DEAs running the application
- Determine applications' desired state, version, and instances and compare to actual state
  - Desire state based upon the Cloud Controller database
- Direct Cloud Controller to take action to correct any discrepancies in the state of applications.

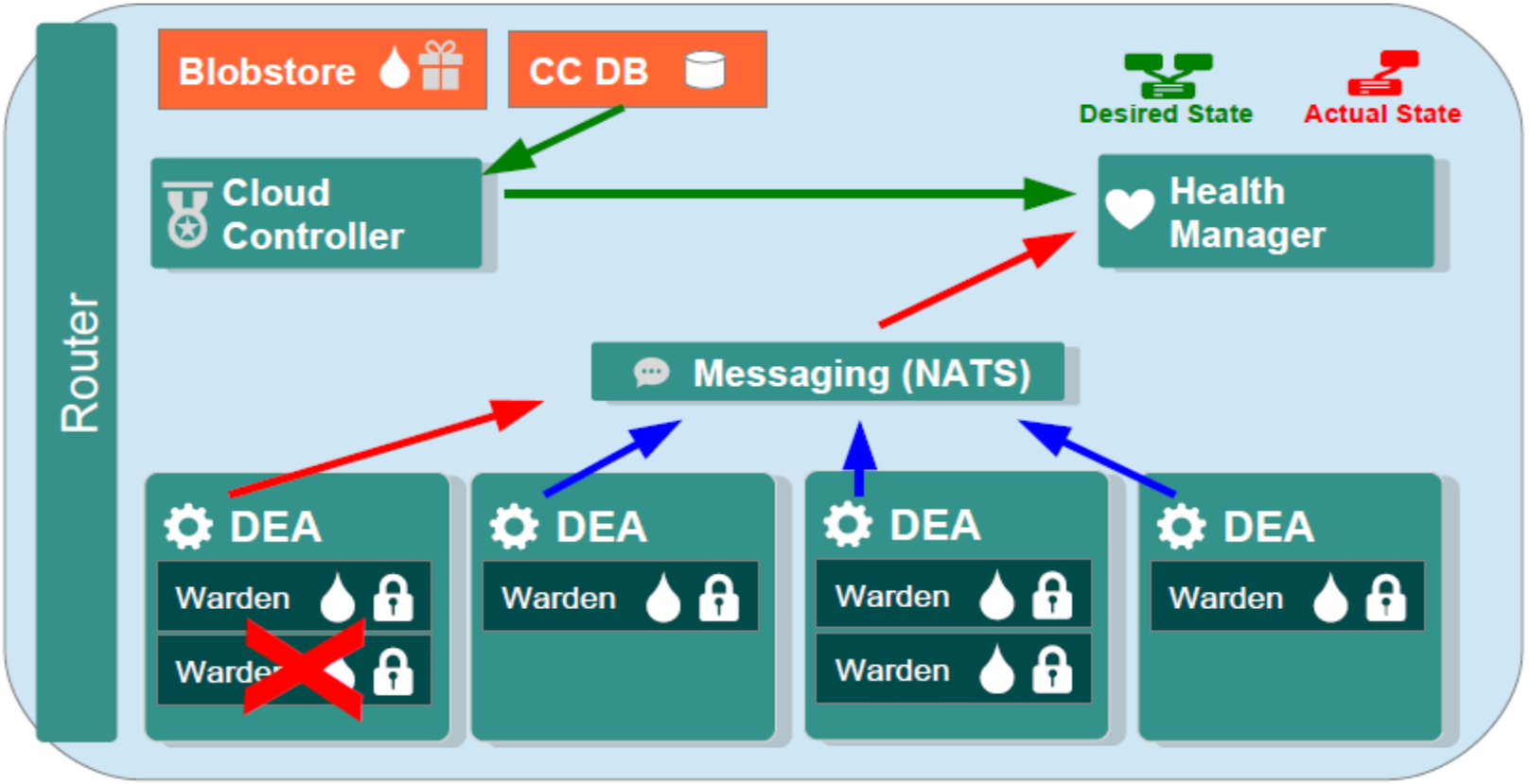
Health Management – how it works?

- Application run within containers on DEAs
- DEAs send heartbeats messages, messages sent to Health Manager
- Health Manager obtains “desired state” from Cloud Controller
- Does “desired state” = “ actual state” ? **Yes**



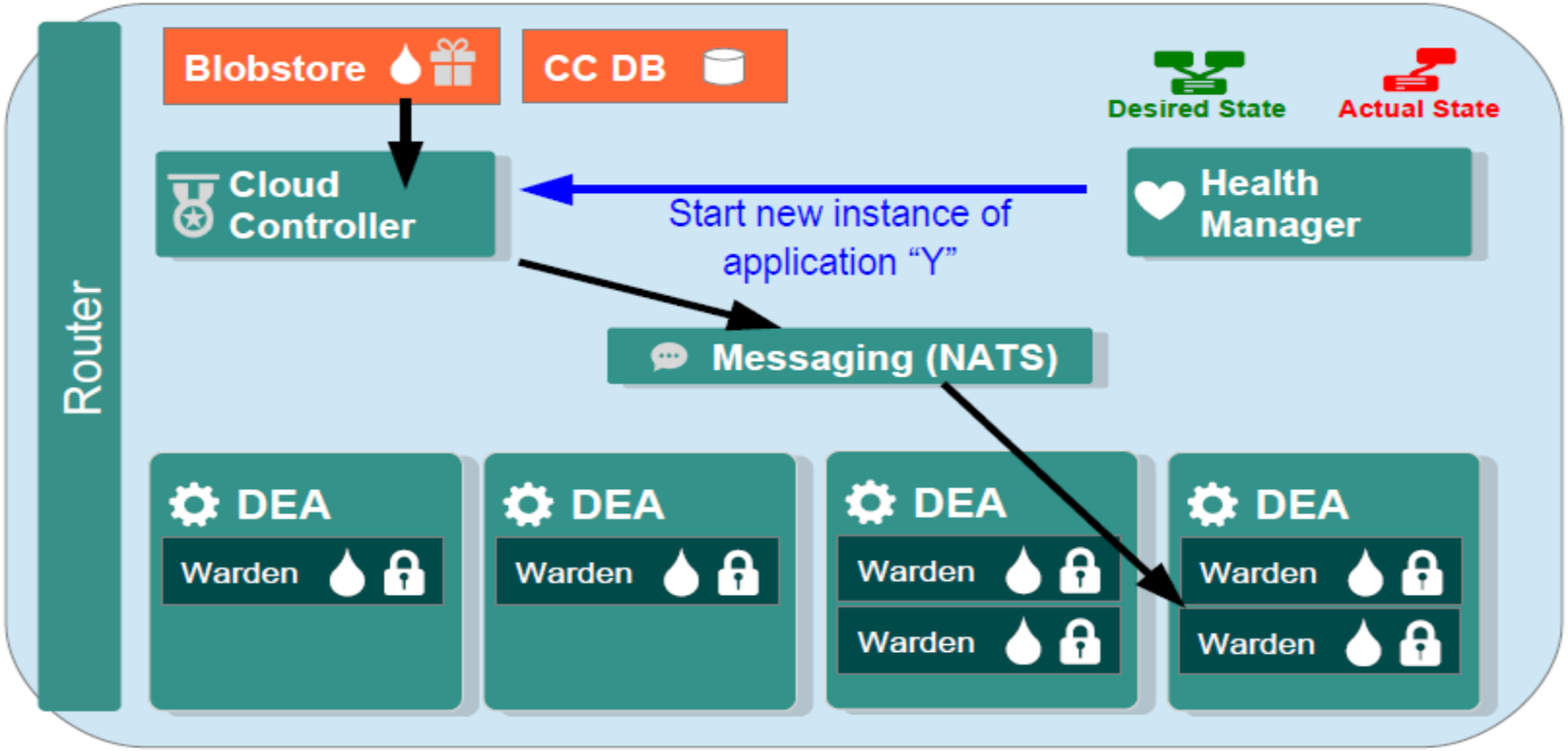
Health Management – Detecting Failures

1. What if an application crashes X?
2. DEAs send heartbeat messages, messages sent to Health Manager
3. Health Manager obtain “desired state” from Cloud Controller
4. “Desired state” = “actual state” ? No



Health Management – Replacing an *Application*

1. Health Manager instructs Cloud Controller
2. Cloud Controller Clones Droplet into container on DEA



Recap

CCDB

Load Balancer

Availability Zone

DEA

Warden

Droplets

HealthManager

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

People matter, results count.

About Capgemini

With more than 130,000 people in 44 countries, Capgemini is one of the world's foremost providers of consulting, technology and outsourcing services. The Group reported 2012 global revenues of EUR 10.3 billion.

Together with its clients, Capgemini creates and delivers business and technology solutions that fit their needs and drive the results they want. A deeply multicultural organization, Capgemini has developed its own way of working, the Collaborative Business Experience™, and draws on Rightshore®, its worldwide delivery model.

www.capgemini.com