High Level Design

This initiative aims to define and realize a new application platform supporting high volumes external and internal channels customer needs (self-management and self-care functionalities).

Business Goals

- Support for ***higher volumes/performance*** of inquiries generated by Mobile App and Dealers platforms while set a standard common data access framework also for other applications/channels
- Promote ***high availability*** and ***"zero downtime"*** for read only data access services
- ***Simplify*** and ***centralize*** service ***operations*** activities

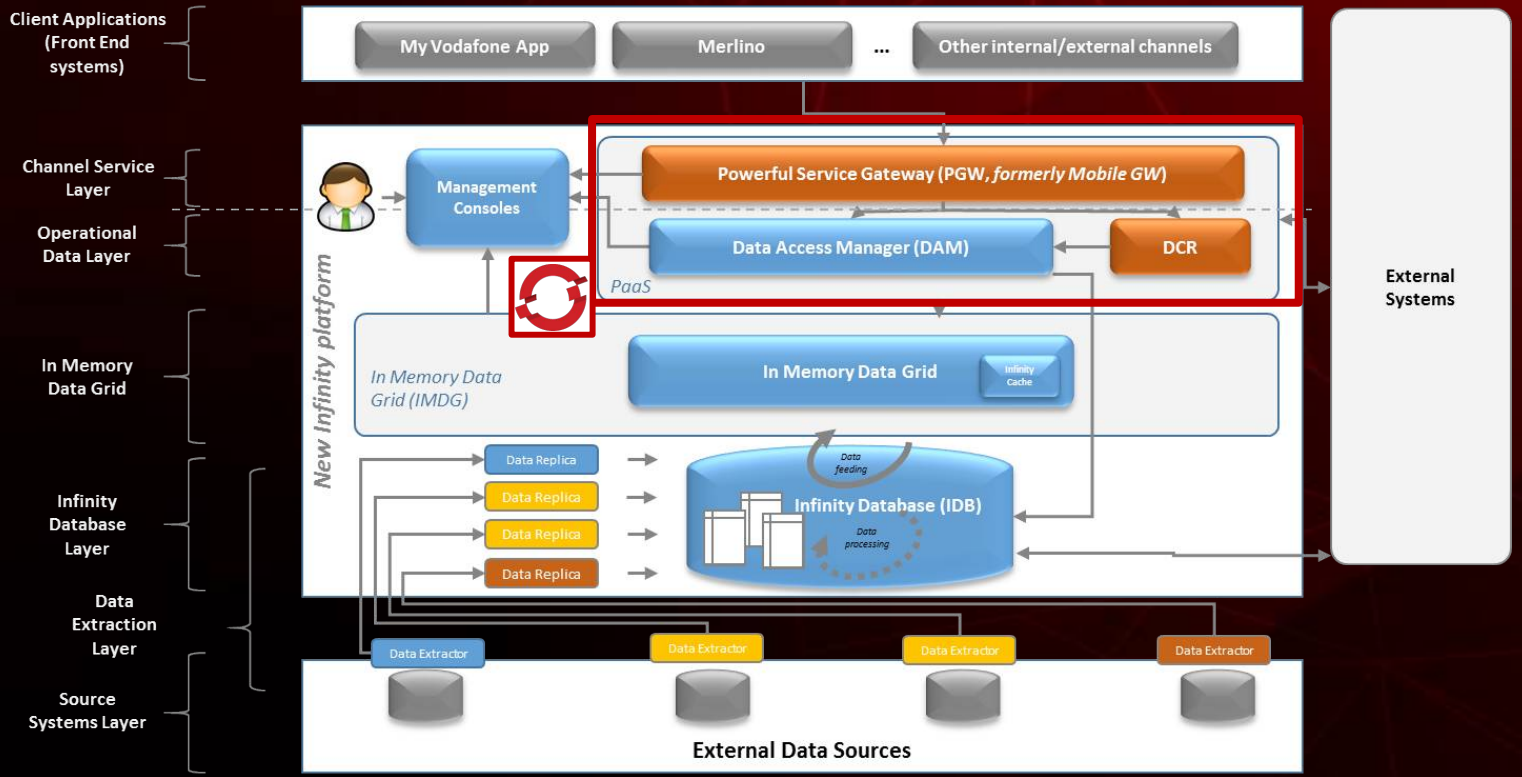
Technical Goals

- ***Scalability***: enable horizontal scalability and overcome potential scale issues in relation to workload generated on operational systems.
- ***Performance***: improve data access performance from external and internal channels (responsiveness, latencies, throughput)
- Operational system ***workload control***: reduce and constantly minimize impacts (and availability dependencies) on data source (back end) systems

In this scenario RedHat OpenShift Enterprise will host custom j2ee applications deployed into JbossEAP cartrige.

Architecture Overview

Following diagram expose an High level overview of the involved components.

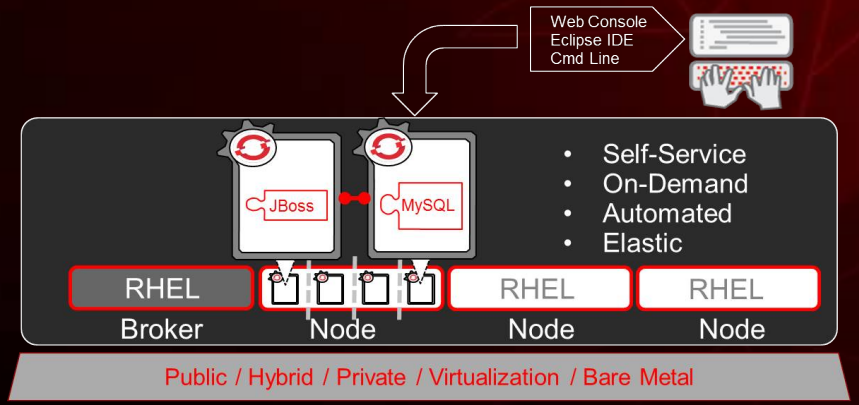


Paas layer is based upon OpenShift Enterprise by RedHat

PaaS Production Deployment Architecture

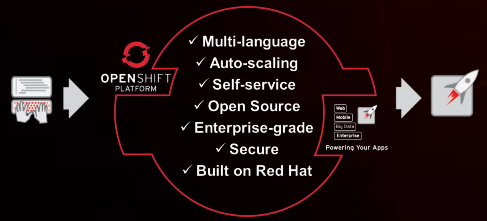
Focus on PaaS Deployment Architecture (1/8)

OpenShift Enterprise architecture overview



All nodes are based on RHEL 6 hosted on Customer virtual DC (vmWare infrastructure)

OSE offer wide, quick and reliable deployment services to host the new generation apps



PaaS is based on OpenShift Enterprise

OSE is build upon 2 classes of roles:

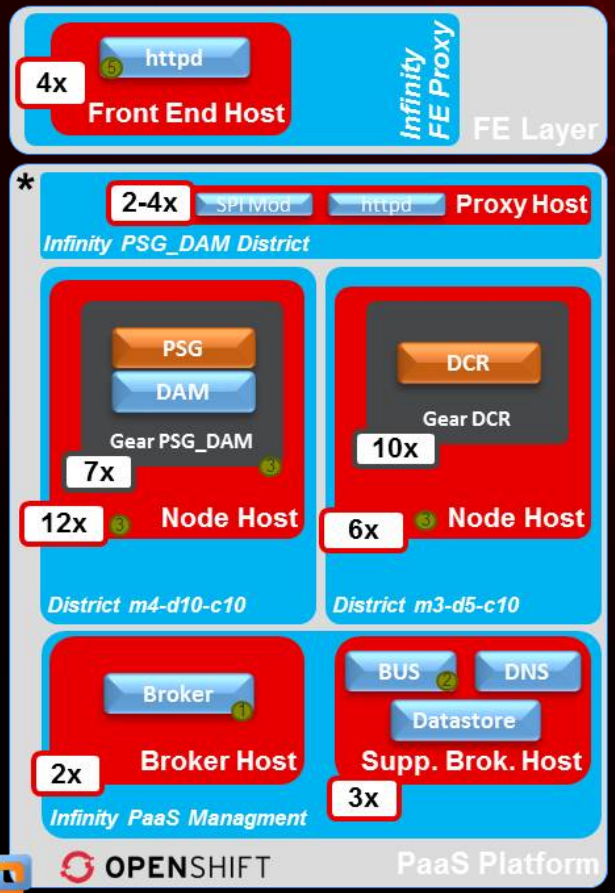
- **OSE Broker** (or Support Broker) Hosts: PaaS Management nodes
- **OSE Node** Hosts: application execution nodes

Every OSE Node is assigned to a district to host gears (one per district).

Gears are hw resources constrained computational entity to deliver, in this scenario:

- PSG & DAM services: J2EE Apps deployed into a jboss EAP container composed to work together to gain better performance by reducing outside calls.
- DCR services: J2EE App deployed into a jboss EAP container leaved on a different district (different kind of resources usage)

Focus on PaaS Deployment Architecture (2/8)

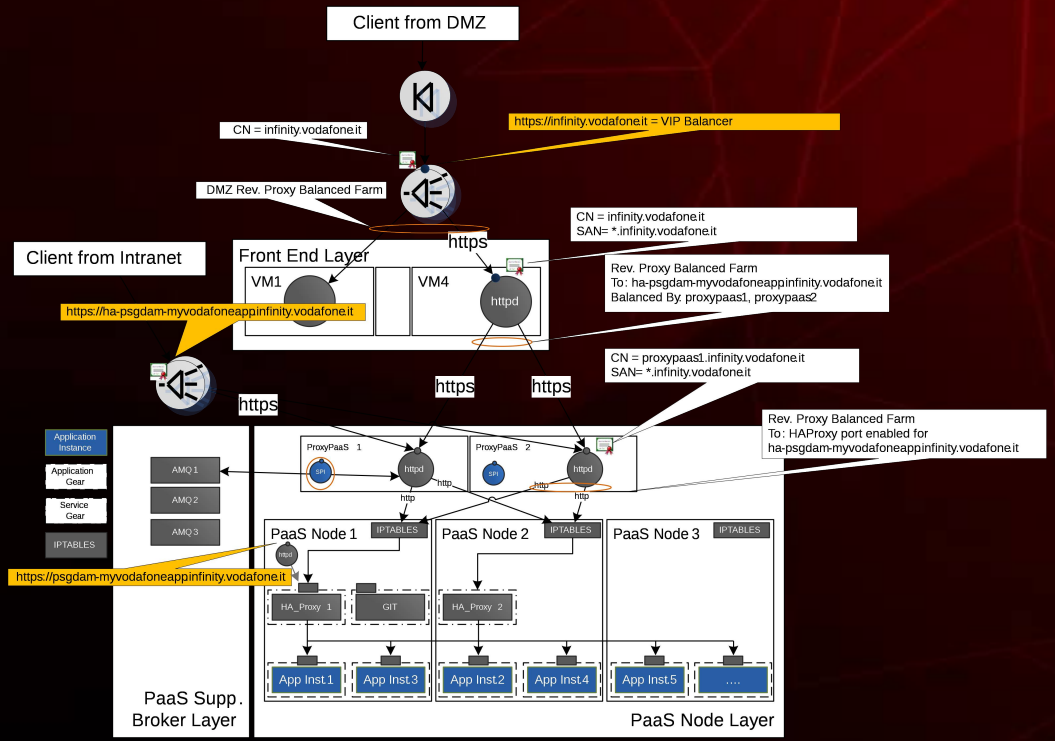


OpenShift Enterprise (OSE) Resource Distribution (Hosts: 4x vCPU, 32GB RAM, 100GB Disk)

- **Brokers**: 2 hosts running OSE broker packages
- **Support Brokers**: 3 hosts running AMQ, Named and MongoDB
- Node District m4-d10-c10: from 2 to 12 hosts running from 1 to 7 gears (defined limit per host). Each gear is configured to gain up to 4GB of RAM and 10GB of Disks
- Node District m3-d5-c10: from 2 to 4 hosts running from 1 to 10 gears (defined limit per host). Each gear is configured to gain up to 3GB of RAM and 5GB of Disks
- Proxy PaaS: from 2 to 4 hosts running httd with SPI module to gain HA BE application
- Front End: 4 hosts statically configured to traverse traffic from DMZ to PaaS

Rif. #	Console Scope
1	[OSE] Application management & gear usage statistics
2	[ActiveMQ] PaaS Message Queue handling
3	[HA_Proxy] Network usage statistics
4	[Custom] Log, RedMode & Config. Management
5	[Apache] http usage statistics

Focus on PaaS Deployment Architecture (3/8)



High Availability for HA Applications

HA applications are enabled on brokers and configured as HA App

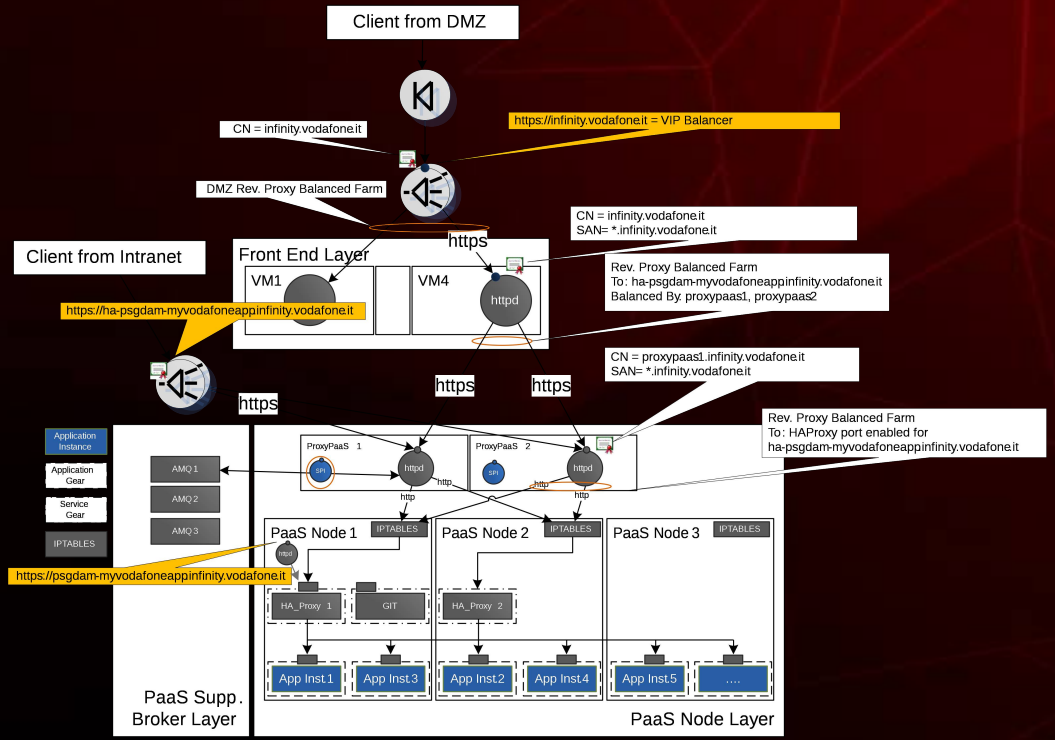
This will enable openshift to bypass any local SPOF (2 or more gears, 2 or more haproxy, 2 or more reverse proxy, imply to use APP URL with “ha-” prefix.

In this deploy scenario, any HA Application is served by a chain of a couple of ProxyPaas (httpd) equipped with SPI module (have a look at custom RPM for details) and 2 HAProxy (see HA scaling configuration for details) to route traffic to BE gears according to failures or administration operations.

Moreover, each proxy farm is served up by a front end balancer to eliminate any SPOF.

Haproxy multiplyer is set to allow 1 haproxy gear per host plus six application gear.

Focus on PaaS Deployment Architecture (4/8)



Zero Downtime deployment

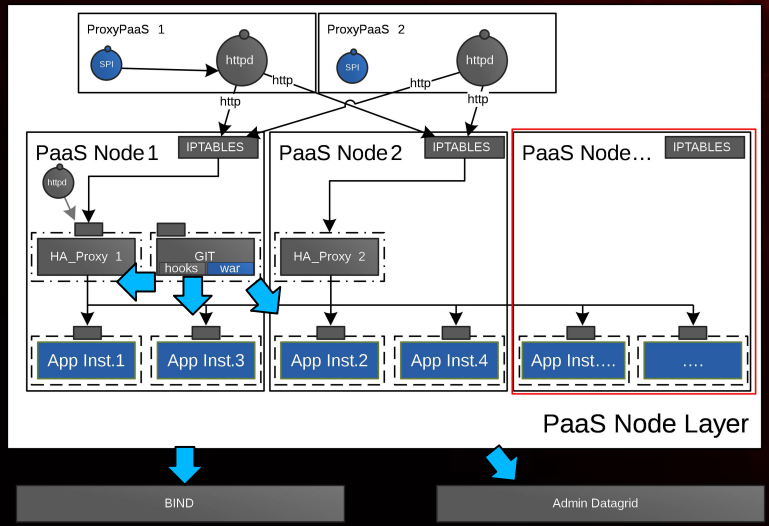
Scaled web cartridge restarts experience zero downtime using multiple HAProxy cartridges (actual infinity **application scale up 1 HAProxy every 7 gears for m4 district** and **1 HAProxy every 10 gears for m3 district** in order to maintain 1 HAProxy for every deployed node).
During deployment openshift restart gears rolling against block of gears, this allow to bypass a full out of order app.

Note: it's required to use "rhc deploy" procedure in conjunction with binary deployments.

Focus on PaaS Deployment Architecture (5/8)

Scalability is based upon OpenShift Enterprise linear scaling feature

PaaS applications are developed to be agnostic about local resources. Configuration are store on environment variables (ported outside war packages), DNS local pointer and datagrid: thos allow packages to be ported across environments and instances enabling linear scalability:



Scalability Note

Starting from a minimum of 2 Hosts and 2 gears each HA application can scale up to a fixed number of gears for each added node.

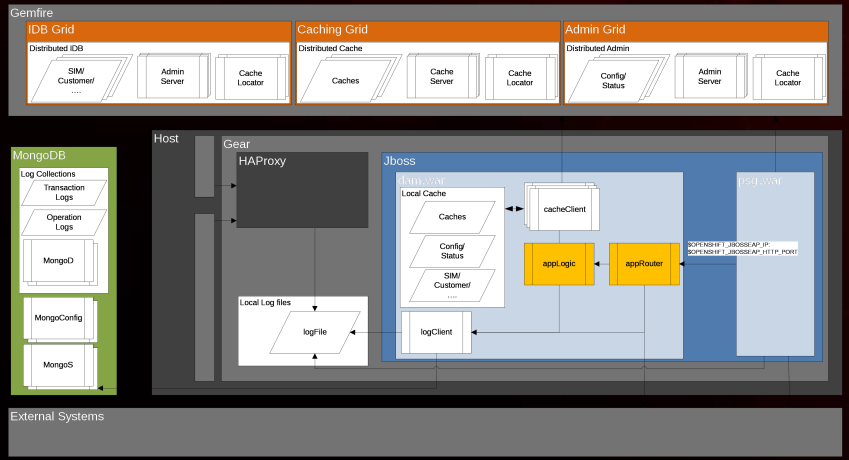
Depending on resourcelimits.conf:

- 7 gears for m4 district
- 10 gears for m3 district

Focus on PaaS Deployment Architecture (6/8)

Gear packages details

Each gear hosts a Jboss EAP as primary application container.
Communication between internal components (different packager residing into same gear) use $OPENSHIFT_JBOSSEAP_IP: $OPENSHIFT_JBOSSEAP_HTTP_PORT environments to address each others.
Other configurations are cached by exported environments variables (see action hooks for details) ore by using a region (Admin) of the in-memory-datagrid. To implement a fire and forget method, logs are pulled by using an asyncronus logger with local and remote appender.



Each gear hosts:

- 1 jboss EAP instance
- 1 or more primary app components (war apps)
- optional modules
- optional action hooks
- optional technical component (at least 1 HAProxy per host this implementation)

Focus on PaaS Deployment Architecture (7/8)

Gear resource distribution

Into a single host, gears resource allocation is controlled by cgroups to limit ram, cpu and disk quota to limit disk usage.
Following tables show details on gears profiles (For each gear the JVM_HEAP_RATIO is settled to 0.8 (80%), see resource_limits.conf for details)

District m4-d10-c10: 2-16 nodes hosting psg.war & dam.war plus 1 haproxy per host, 80% of the gear memory is reserved for jboss

Host m4-d10-c10 District (psgdam)	
vCPU [#]	4
RAM [GB]	32
Disk [GB]	150
TOT Net Bandwith [gbit/s]	10

OS Resources reservation	
vCPU [%]	20%
RAM [GB]	2
Disk	50
Net Bandwith [%]	20%

Spare Resources @ Max alloc	
RAM [GB]	2
Disk [GB]	30

Gear Resources Allocation	
Gear Name	m4-d10-c10
Storage Partition [GB]	10
Memory Limit [GB]	4
MAX CPU% per gear	11%

Other Gears Configuration Params	
max_active_gears	7
no_overcommit_active	true
limits_nproc	1.500
quota_blocks	10.485.760
memory_limit_in_bytes	4.294.967.296
tc_max_bandwidth [mbit/s]	8.000
cpu_shares	320

District m3-d5-c10: 2-4 nodes hosting dcr.war plus 1 haproxy every 2 gears, 80% of the gear memory is reserved for jboss

Host m3-d5-c10 District (dcr)	
vCPU [#]	4
RAM [GB]	32
Disk [GB]	150
TOT Net Bandwith [gbit/s]	10

OS Resources reservation	
vCPU [%]	20%
RAM [GB]	2
Disk	50
Net Bandwith [%]	20%

Spare Resources @ Max alloc	
RAM [GB]	0
Disk [GB]	50

Gear Resources Allocation	
Gear Name	m3-d5-c10
Storage Partition [GB]	5
Memory Limit [GB]	3
MAX CPU% per gear	11%

Other Gears Configuration Params	
max_active_gears	10
no_overcommit_active	true
limits_nproc	1.500
quota_blocks	5.242.880
memory_limit_in_bytes	3.221.225.472
tc_max_bandwidth [mbit/s]	8.000
cpu_shares	320

PaaS Nodes details

Hostname	ose-n-p-1 to ose-n-p-2
Env	PROD
Server logical name	ProxyPaas
Location	vmWare Cluster1 odd vmWare Cluster2 even
Components	Httpd, SPI
Role	Paas Proxy
OS	Red Hat 6.5 Enterprise Edition
Package	Httpd, SPI
CORE/vCPU	4
RAM (GB)	32
Volumes (GB)	100

Hostname	ose-b-p-1 to ose-b-p-2
Env	PROD
Server logical name	PaaS Broker 1 to 2
Location	vmWare Cluster1 per i nodi dispari vmWare Cluster2 per i nodi pari
Components	-
Role	Paas Broker
OS	Red Hat 6.5 Enterprise Edition
Package	Open Shift Enterprise v 2.1
CORE/vCPU	4
RAM (GB)	32
Volumes (GB)	150

Hostname	ose-n-p-3 to ose-n-p-20
Env	PROD
Server logical name	PaaS Node 3-20
Location	vmWare Cluster1 odd vmWare Cluster2 even
Components	PWG DAM DCR
Role	Paas Nodes
OS	Red Hat 6.5 Enterprise Edition
Package	Open Shift Enterprise v 2.1 JbossEAP cartridge
CORE/vCPU	4
RAM (GB)	32
Volumes (GB)	150

Hostname	ose-sb-p-1 to ose-sb-p-3
Env	PROD
Server logical name	PaaS Supp. Broker 1 to 3
Location	vmWare Cluster1 odd vmWare Cluster2 even
Components	-
Role	Paas Supp. Broker
OS	Red Hat 6.5 Enterprise Edition
Package	Named AMQ Mongo
CORE/vCPU	4
RAM (GB)	32
Volumes (GB)	150


Thank You