

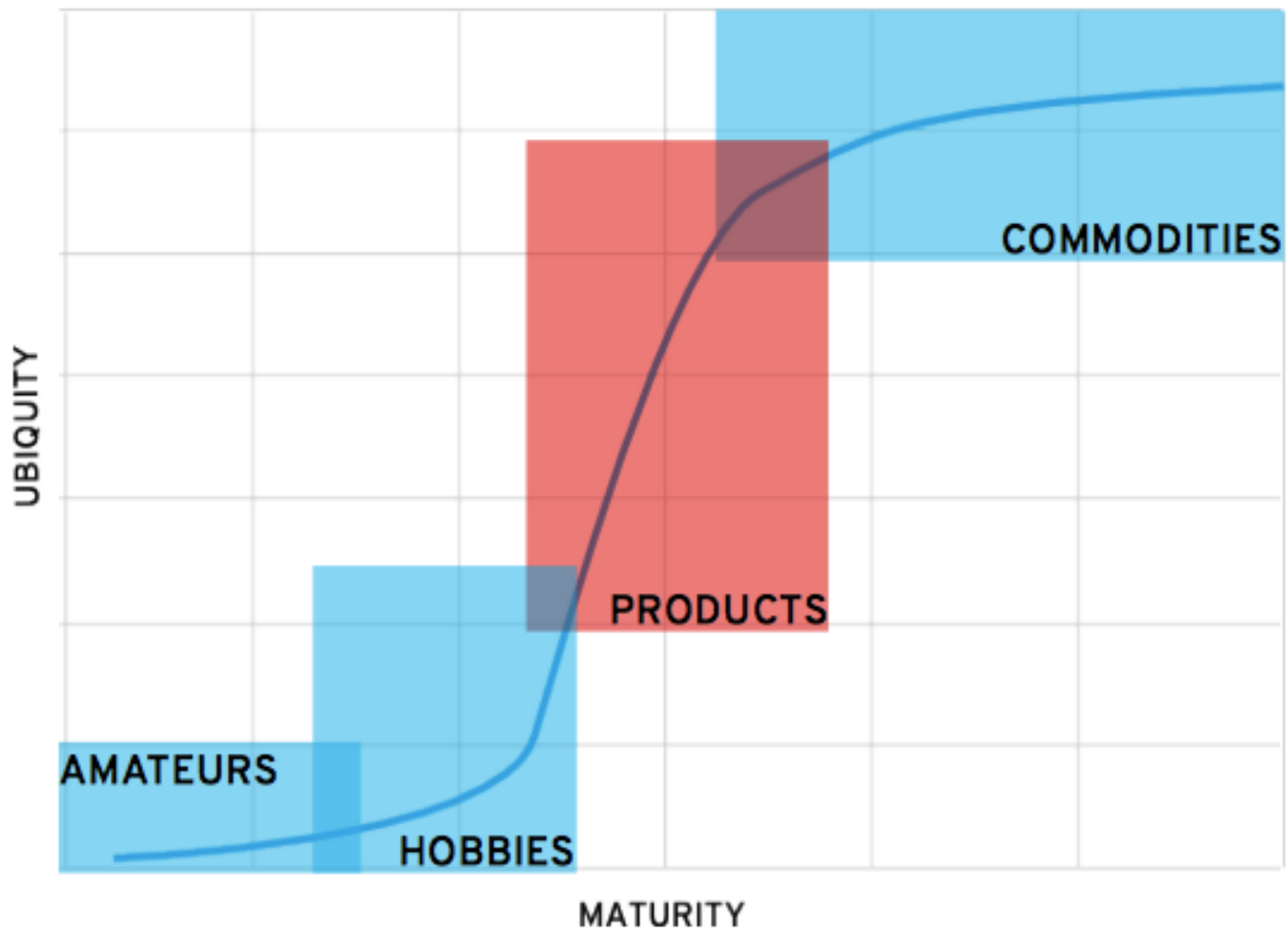
Manufacturing Polkadot Warplanes

Shawn Wells | shawn@redhat.com









The background of the slide is a faded, light blue financial chart. It features a grid with two main data series: a line graph at the top and a bar chart at the bottom. The line graph has several peaks and troughs, with numerical values like 1279.2, 1319.1, and 1327.9 visible. The bar chart at the bottom shows a series of vertical bars of varying heights, generally trending upwards from left to right. The overall aesthetic is professional and data-oriented.

JPMorgan's plumbing problem



ENTER

The NSA's security challenge

Password:

• • • • • • • • • • |



IaaS

PaaS

SaaS

APPLICATION

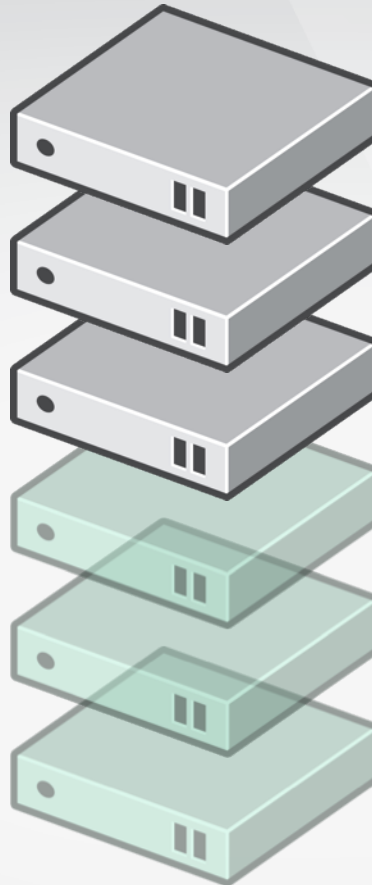
APPLICATION PLATFORM
(JBoss, PHP, RUBY, ETC)

OPERATING SYSTEM
(RHEL)

VIRTUALIZATION
(RHEV)

HARDWARE
(x86)

STORAGE
(RHS)



Managed and
Controlled by
Customer



Provider Managed

Increased Control

Increased Automation

IaaS

PaaS

SaaS

APPLICATION

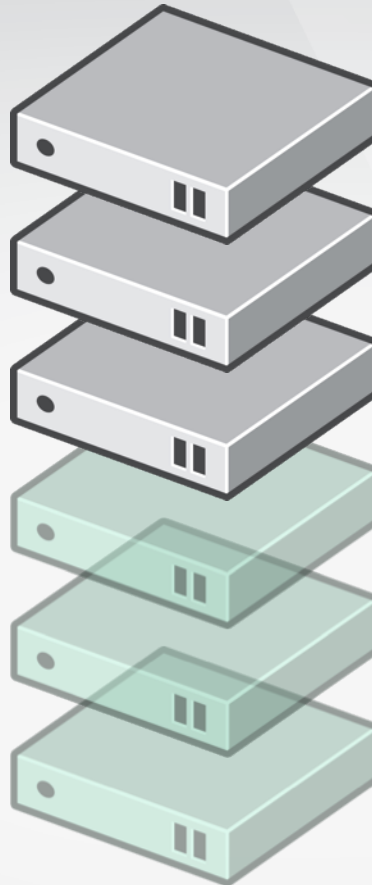
APPLICATION PLATFORM
(JBoss, PHP, RUBY, ETC)

OPERATING SYSTEM
(RHEL)

VIRTUALIZATION
(RHEV)

HARDWARE
(x86)

STORAGE
(RHS)



Managed and
Controlled by
Customer



Provider Managed

Increased Control

Increased Automation

IaaS

PaaS

SaaS

APPLICATION

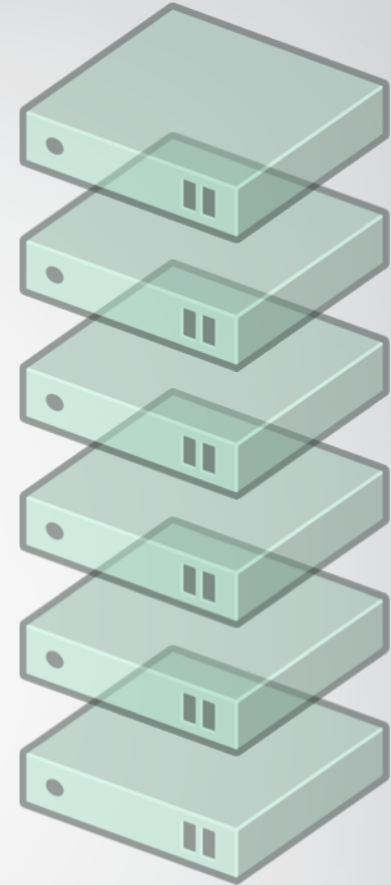
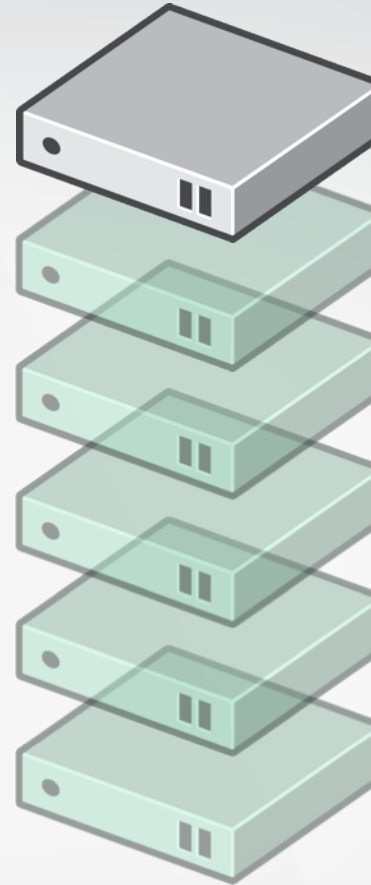
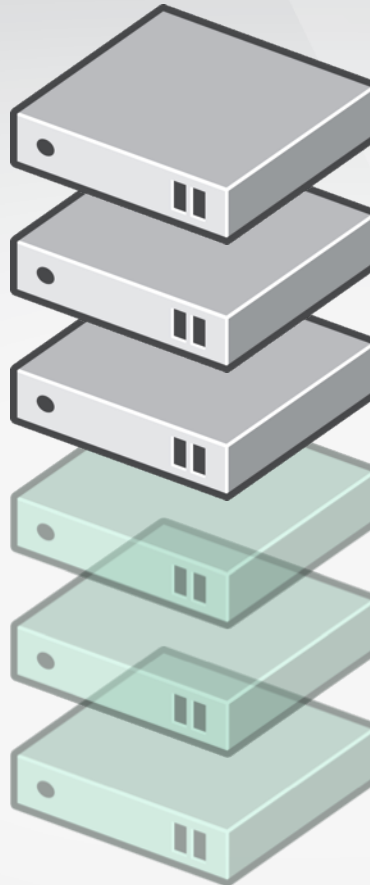
APPLICATION PLATFORM
(JBoss, PHP, RUBY, ETC)

OPERATING SYSTEM
(RHEL)

VIRTUALIZATION
(RHEV)

HARDWARE
(x86)

STORAGE
(RHS)



Managed and
Controlled by
Customer



Provider Managed

Increased Control

Increased Automation

CRAFTWORK → MASS PRODUCTION

Physical

Virtualized

With PaaS

How to Build an App:

1. Have Idea
2. Get Budget
3. Submit hardware acquisition request
4. Wait
5. Get Hardware
6. Rack and Stack Hardware
7. Install Operating System
8. Install Operating System Patches/Fix-Packs
9. Create user Accounts
10. Deploy framework/appserver
11. Deploy testing tools
12. Code
13. Test
14. Configure Prod servers (and buy them if needed)
15. Push to Prod
16. Launch
17. Order more servers to meet demand
18. Wait...

CRAFTWORK → MASS PRODUCTION

Physical

How to Build an App:

1. Have Idea
2. Get Budget
3. Submit hardware acquisition request
4. Wait
5. Get Hardware
6. Rack and Stack Hardware
7. Install Operating System
8. Install Operating System Patches/Fix-Packs
9. Create user Accounts
10. Deploy framework/appserver
11. Deploy testing tools
12. Code
13. Test
14. Configure Prod servers (and buy them if needed)
15. Push to Prod
16. Launch
17. Order more servers to meet demand
18. Wait...

Virtualized

How to Build an App:

1. Have Idea
2. Get Budget
3. Submit VM Request request
4. Wait
5. Deploy framework/appserver
6. Deploy testing tools
7. Code
8. Test
9. Configure Prod VMs
10. Push to Prod
11. Launch
12. Request More Prod VMs to meet demand
13. Wait
14. Deploy app to new VMs
15. Etc.

With PaaS

CRAFTWORK → MASS PRODUCTION

Physical

How to Build an App:

1. Have Idea
2. Get Budget
3. Submit hardware acquisition request
4. Wait
5. Get Hardware
6. Rack and Stack Hardware
7. Install Operating System
8. Install Operating System Patches/Fix-Packs
9. Create user Accounts
10. Deploy framework/appserver
11. Deploy testing tools
12. Code
13. Test
14. Configure Prod servers (and buy them if needed)
15. Push to Prod
16. Launch
17. Order more servers to meet demand
18. Wait...

Virtualized

How to Build an App:

1. Have Idea
2. Get Budget
3. Submit VM Request request
4. Wait
5. Deploy framework/appserver
6. Deploy testing tools
7. Code
8. Test
9. Configure Prod VMs
10. Push to Prod
11. Launch
12. Request More Prod VMs to meet demand
13. Wait
14. Deploy app to new VMs
15. Etc.

With PaaS

How to Build an App:

1. **Have Idea**
2. **Get Budget**
3. **Code**
4. **Test**
5. **Launch**
6. **Automatically Scale**

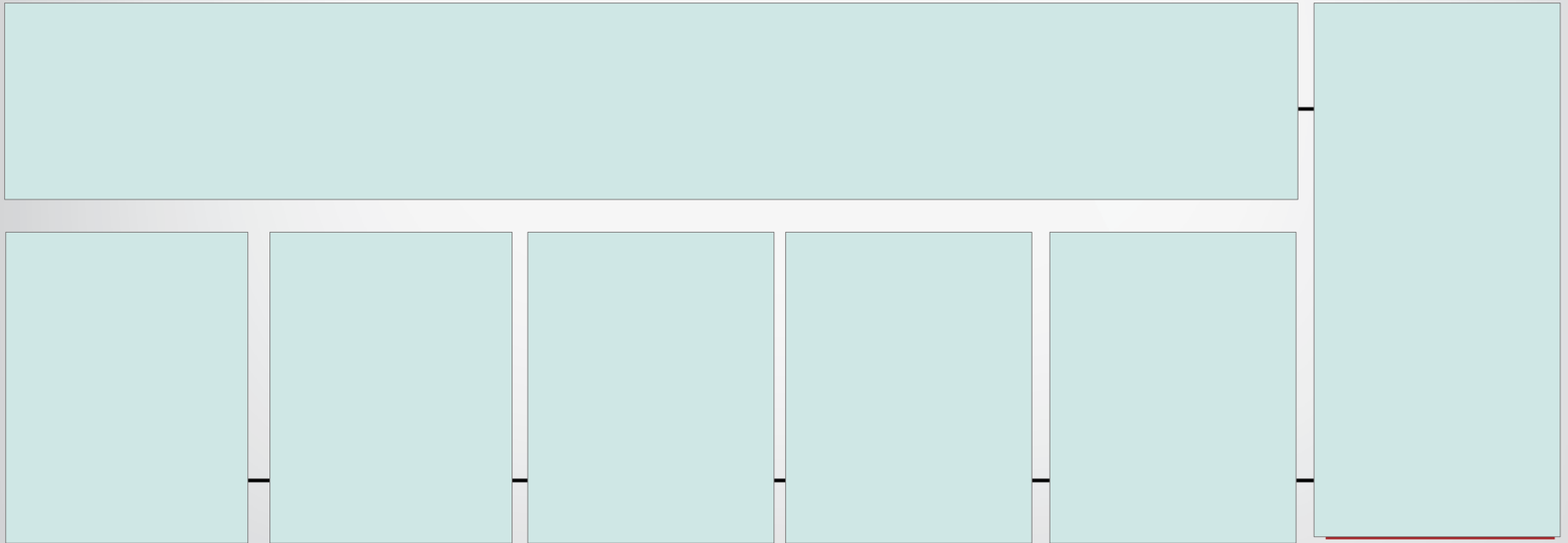


MANUFACTURING FOUNDATIONS

(for IaaS)

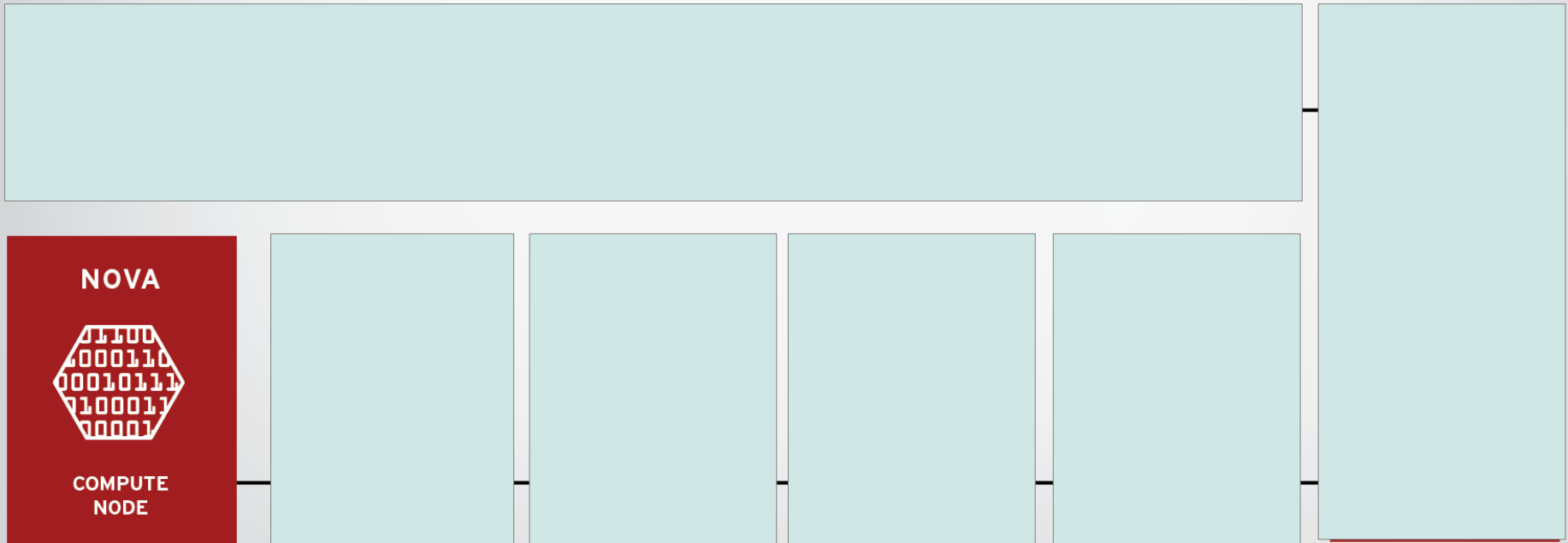
RED HAT OPENSTACK

- MODULAR ARCHITECTURE
- DESIGNED TO SCALE OUT
- BASED ON (GROWING) SET OF CORE SERVICES



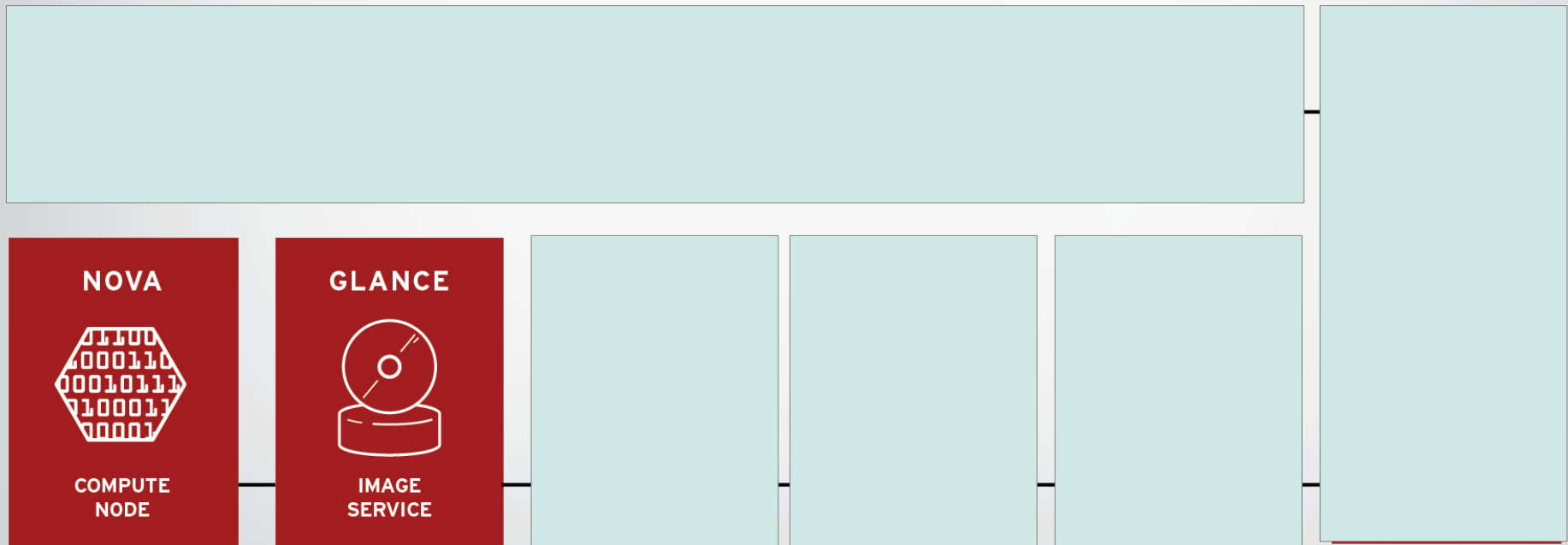
RED HAT OPENSTACK

- MODULAR ARCHITECTURE
- DESIGNED TO SCALE OUT
- BASED ON (GROWING) SET OF CORE SERVICES



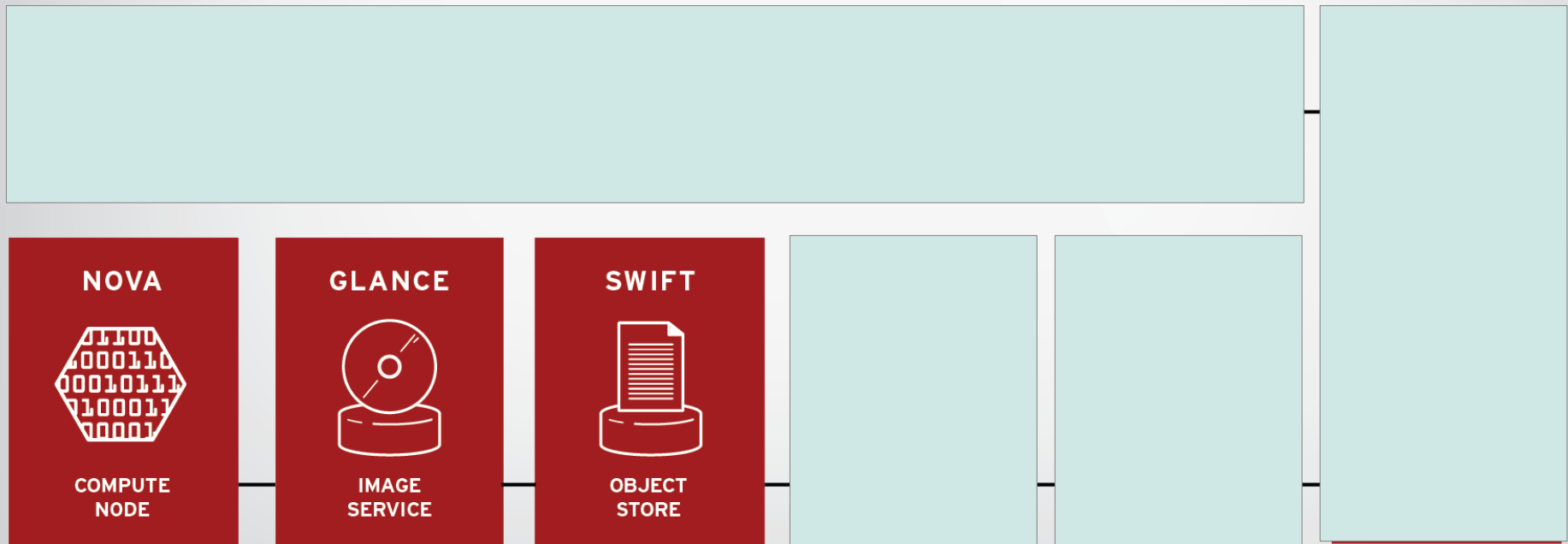
RED HAT OPENSTACK

- MODULAR ARCHITECTURE
- DESIGNED TO SCALE OUT
- BASED ON (GROWING) SET OF CORE SERVICES



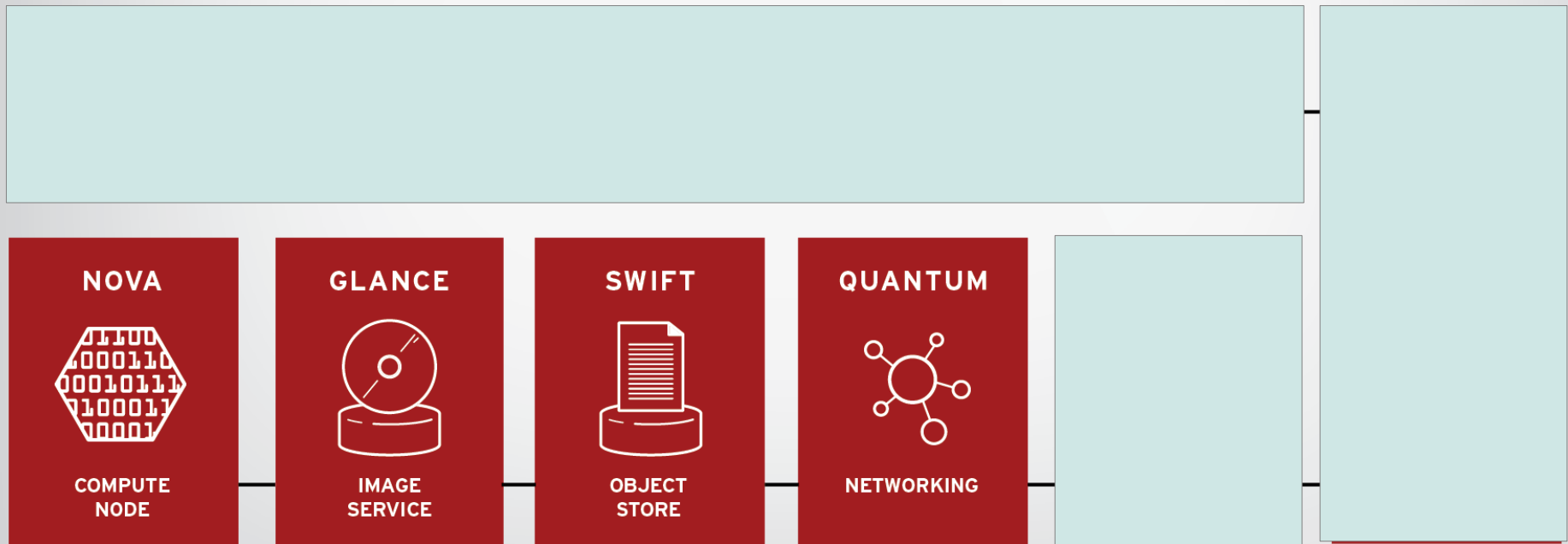
RED HAT OPENSTACK

- MODULAR ARCHITECTURE
- DESIGNED TO SCALE OUT
- BASED ON (GROWING) SET OF CORE SERVICES



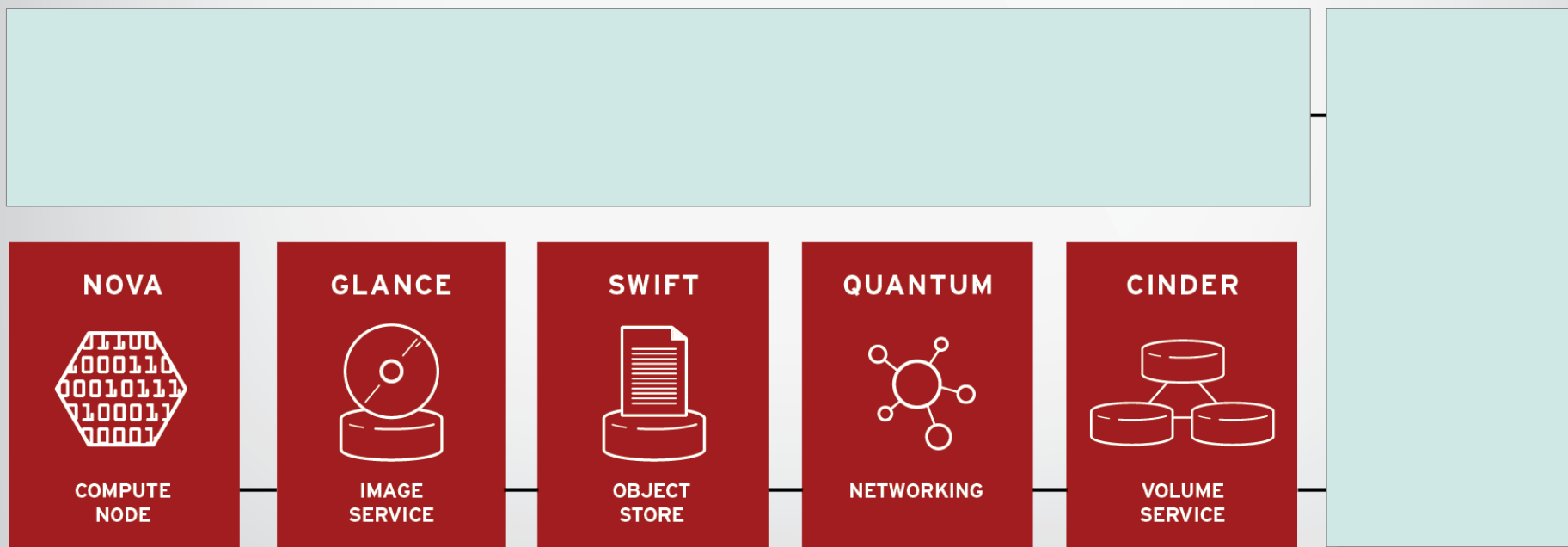
RED HAT OPENSTACK

- MODULAR ARCHITECTURE
- DESIGNED TO SCALE OUT
- BASED ON (GROWING) SET OF CORE SERVICES



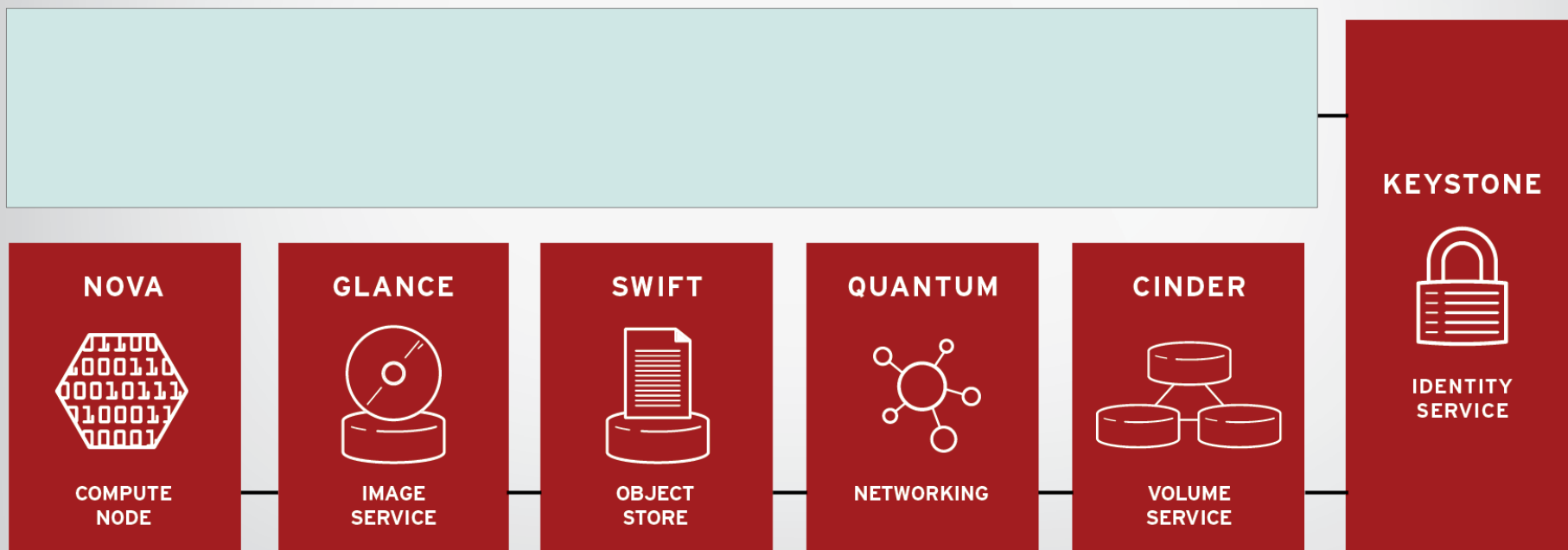
RED HAT OPENSTACK

- MODULAR ARCHITECTURE
- DESIGNED TO SCALE OUT
- BASED ON (GROWING) SET OF CORE SERVICES



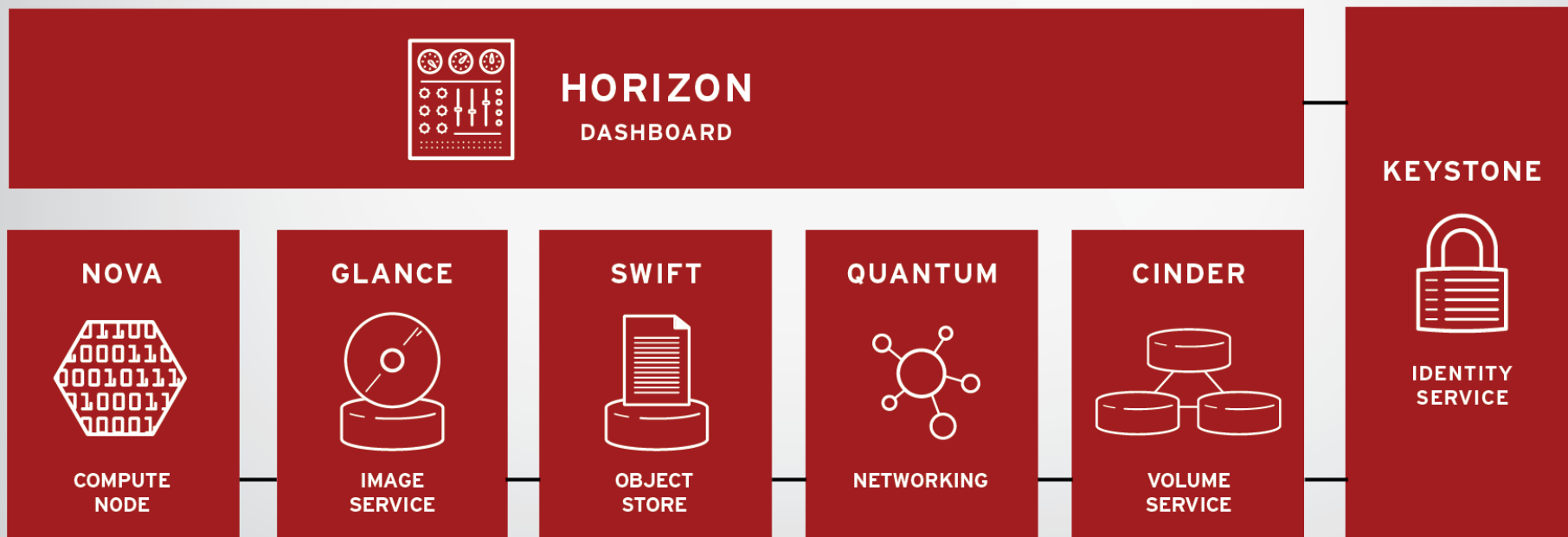
RED HAT OPENSTACK

- MODULAR ARCHITECTURE
- DESIGNED TO SCALE OUT
- BASED ON (GROWING) SET OF CORE SERVICES



RED HAT OPENSTACK

- MODULAR ARCHITECTURE
- DESIGNED TO SCALE OUT
- BASED ON (GROWING) SET OF CORE SERVICES



RED HAT INVOLVEMENT

- Be to OpenStack what Red Hat is to Linux
- Create Enterprise Distribution
 - * Supportability
 - * Stability
 - * Enterprise Features (e.g. security, performance)
 - * Certified Ecosystem
 - * Lifecycle

OPENSTACK RELEASE CADENCE

- Upstream
 - * Source code only
 - * Releases every 6 months
 - * No bugfixes / snapshots after next release
- RDO (e.g. “Fedora” of OpenStack)
 - * Follows upstream cadence
 - * Delivers binaries

OPENSTACK RELEASE CADENCE

- Red Hat OpenStack
 - * Initially 1 year lifecycle
(e.g. support for Folsom ends after Grizzly)
 - * Will increase lifecycle over time
(likely to move to 2 years)
 - * Ecosystem of Support
 - * Government Ready

SOLUTION ENABLEMENT: CORNERSTONE

CORNERSTONE

Open, unified, and extensible scale-out object storage solution for on-premise, virtualized and cloud environments

Which includes

Pervasive, flexible encryption with an innovative approach to unified symmetrical key management

CORNERSTONE



FILES AND FOLDERS

Documents, Photos,
Videos, and Images

10101110101010101
10101011001010111
1010111010001010
0101101001010101

BIG DATA

Log files, RFID data



OBJECTS

Long Tail Data



CORNERSTONE

**DEPLOY
ANYWHERE**

Datacenter, Private Cloud,
Public Cloud

**DRASTICALLY
LOWER COST**

You Choose the Infrastructure -
x86, Virtual

**FOUNDATION
FOR BIG DATA**

Scale-out NAS and Object,
Unstructured Data

**OPEN AND
EXTENSIBLE**

Create Extensible Modules,
Open Source

PANDORA RADIO



- 1.2 PB of audio served per week
- 13 million files
- 80 Storage Nodes
- Over 50 GB/sec peak traffic

Challenges

- Explosive user growth
- 12 formats per song
- 'Hot' Content

Solution

- 3 data centers, 6 nodes per
- Replication between centers
- 2PB+ Capacity

法人総合トップ

サービス・ソリューション

キャンペーン・イベント情報

導入事例

サポート・障害情報

ICT Vision

法人のお客さま総合 > カテゴリーからサービスを探す > データセンター・ホスティング > ストレージホスティング > Bizシンプルディスク
ボリュームタイプ

Bizシンプルディスク ボリュームタイプ

2011年8月3日
提供開始！

「Bizシンプルディスク ボリュームタイプ」は安心・安全なNTTComのデータセンターを利用し、大容量仮想ハードディスクを低価格で提供するサービスです。
VPNとの直結による自社と同等のセキュリティ環境で、ペタバイトクラスの大容量のデータを保管できます。



お問い合わせ／お申し込み
資料請求

お電話

サービスのお問い合わせ・お見積もり
法人コンタクトセンター
0120-106107
受付時間 9:30～17:00（土日休日を除く）

**Bizシンプルディスク ボ
リュームタイプトップ**

・サービス詳細

・利用シーン

・料金

・関連サービス

Bizシンプルディスク ボリュームタイプ

Community Driven Innovation ードディスクサービス

NTT Communications'
"Biz Simple Disk"

Multi-tenant storage service
10TB – 3PB per customer
99.99999999% uptime

DESIGN CONSIDERATIONS

- Consistent, global information accessibility through REST API
- Consistent access through high and low-latency networks
- Cost effectiveness and scalability for big data growth
- Open system, empowering developers

DESIGN CONSIDERATIONS

INFRASTRUCTURE

- Integration with PKI (NSA CASPORT), Active Directory, LDAP

-

-

DESIGN CONSIDERATIONS

INFRASTRUCTURE

- Integration with PKI (NSA CASPORT), Active Directory, LDAP
- Data retains encryption across backup and disaster recovery
-

DESIGN CONSIDERATIONS

INFRASTRUCTURE

- Integration with PKI (NSA CASPORT), Active Directory, LDAP
- Data retains encryption across backup and disaster recovery
- Encrypted files do not require significant additional space; storage requirements change non perceptibly

DESIGN CONSIDERATIONS

ADMINISTRATION

- Role Based Access Control (RBAC), augmented with Mandatory (MAC) and discretionary (DAC) granularity
- Real-time modification to access control policies, to include time limits, number of accesses, etc
- All operations, including key access, are audited

DESIGN CONSIDERATIONS

END-USER

- De-identify sources & methods: Expand data encryption options to protecting relationships
- Anonymize Datasets: Replace sensitive portions of data records with cryptographic pseudonyms
- Protect and control access to multiple data sources from a query device (e.g. remote wireless tablet vs physical desk at DIA)

SECURITY

- **FIPS VALIDATION**

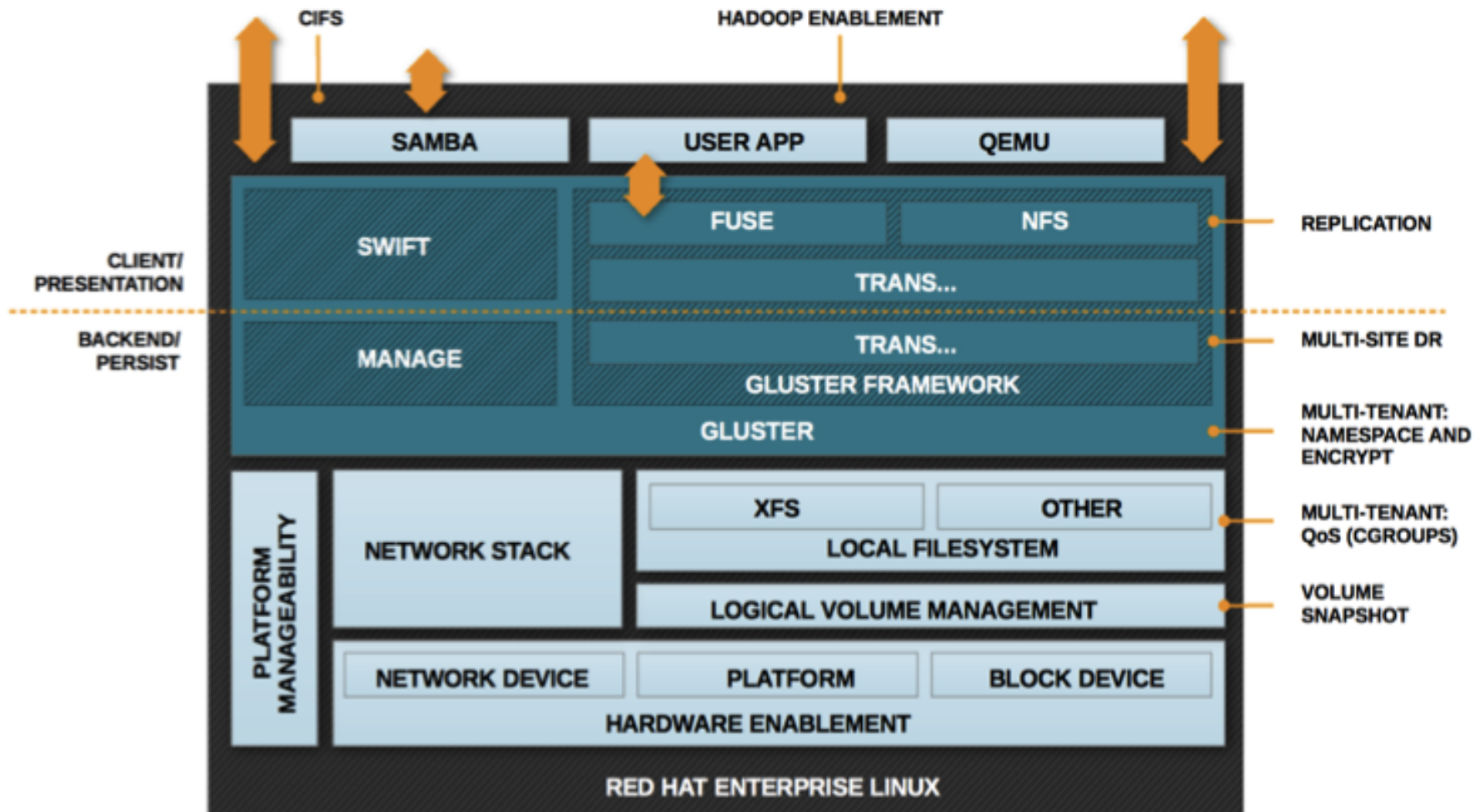
- * Federal Information Processing Standard 140-2

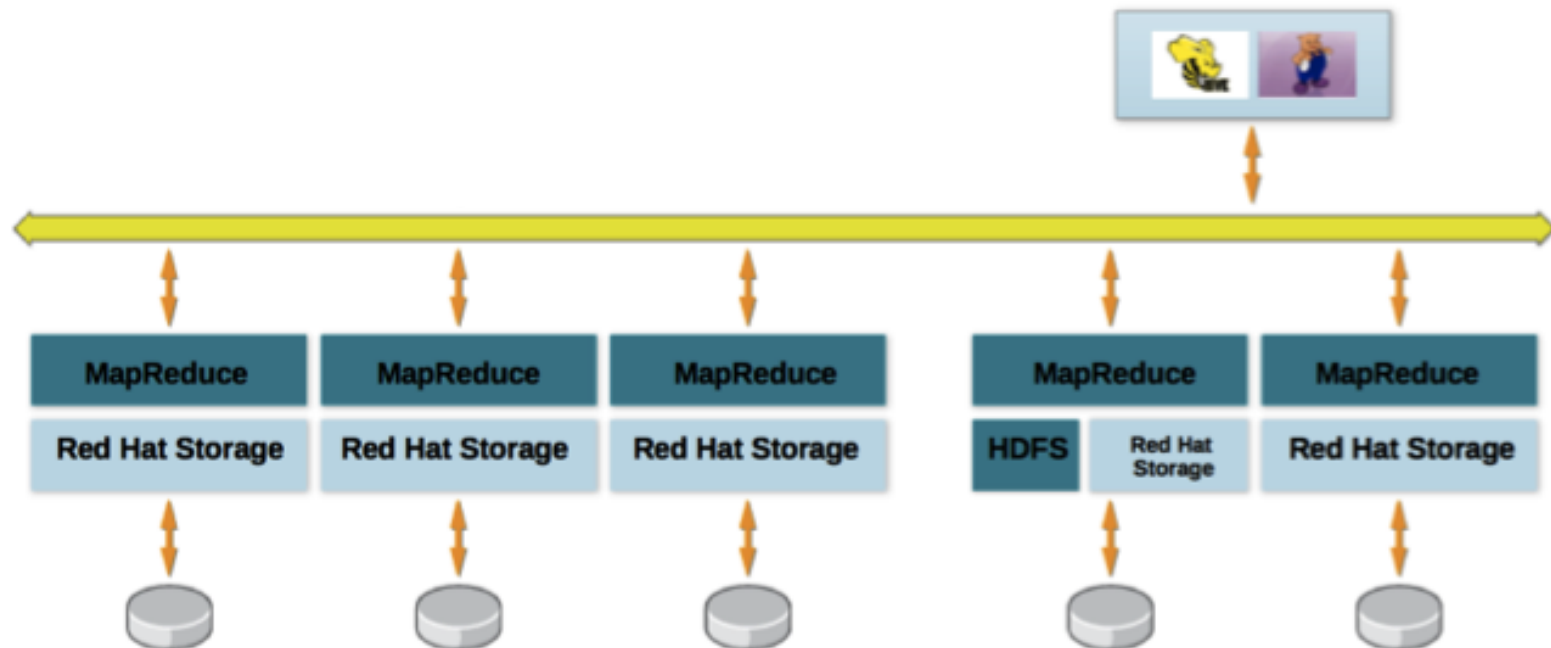
- **COMMON CRITERIA CERTIFIED FOUNDATIONS**

- * Operating System: EAL4+, CAPP/RBAC/LSPP
 - * Crypto: EAL2+

- **MULTIPLE ALGORITHM SUPPORT**

- * AES (256 bits, 196 bits, 128 bits)
 - * TripleDES (168)
 - * SHA-2 hash functions (SHA-224, -256, -384, -512)
 - * ECC and RSA TLS protocols

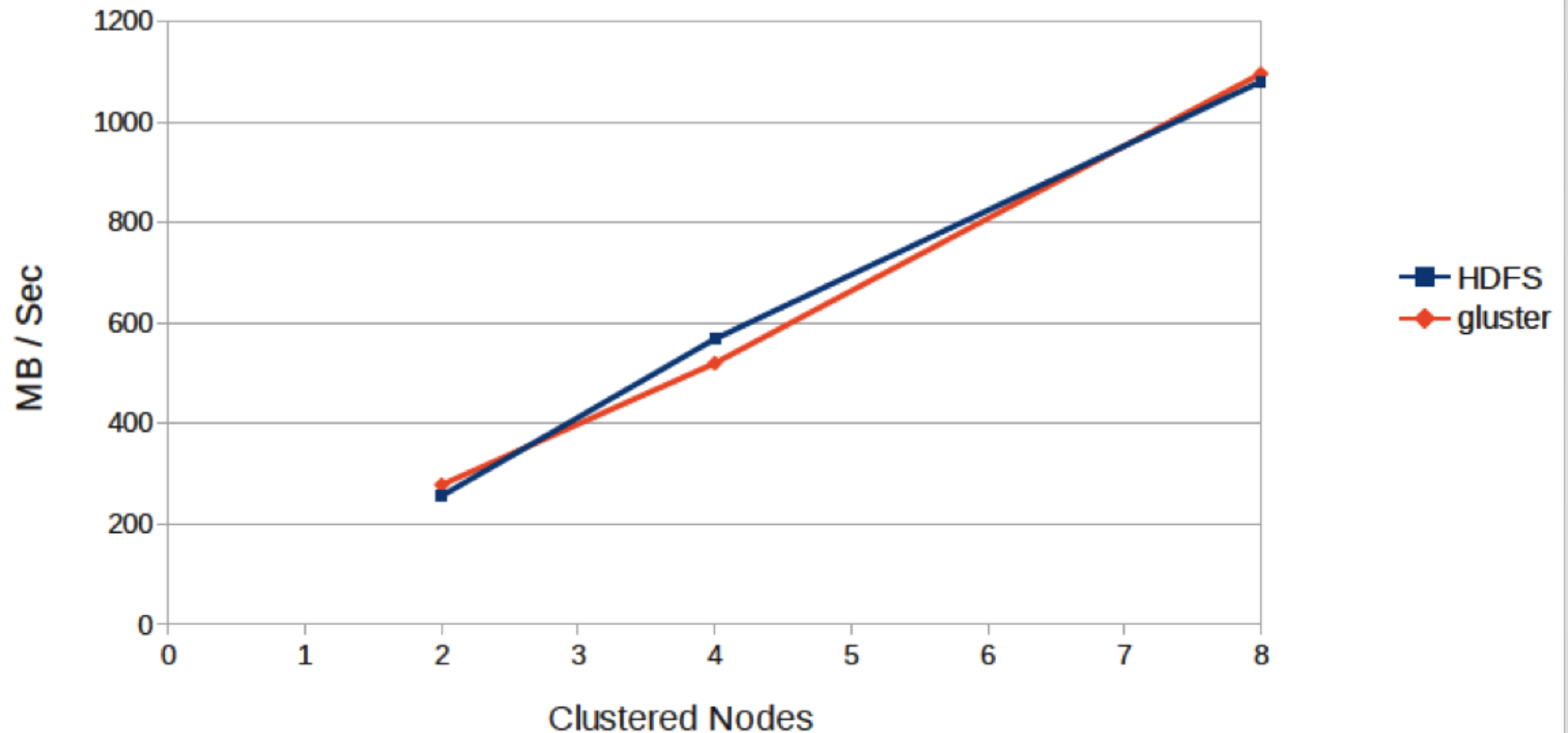




- Out-of-the-box compatibility
- Dramatically increases scale-out capabilities
- Enables Hadoop analytics for unstructured data
- Eliminates HDFS NameNode bottlenecks

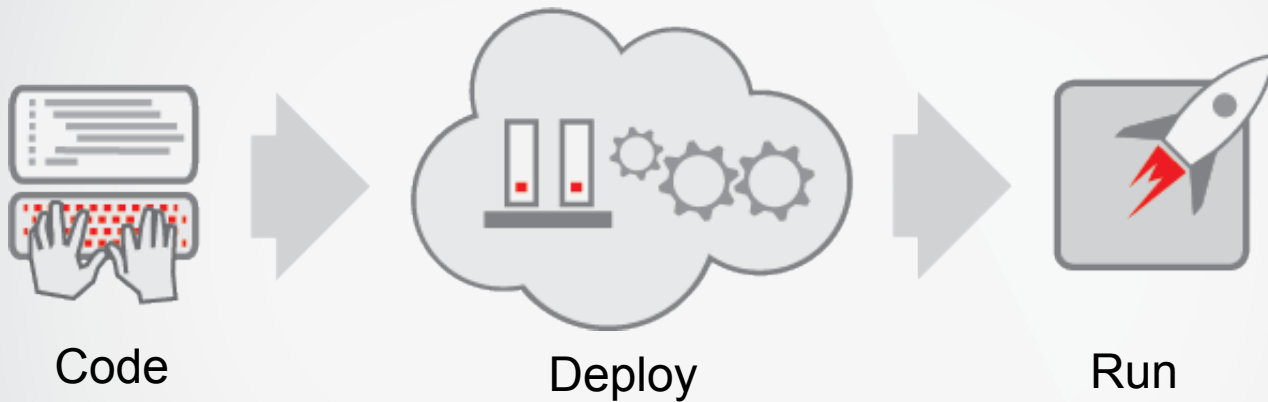
- Compute and storage run on the same systems
- HDFS designed for semi-structured log file data
- Requires data to be resized to be processed by Hadoop and MapReduce

HDFS vs CORNERSTONE



MANUFACTURING FOUNDATIONS **(for PaaS)**

PaaS = Platform as a Service



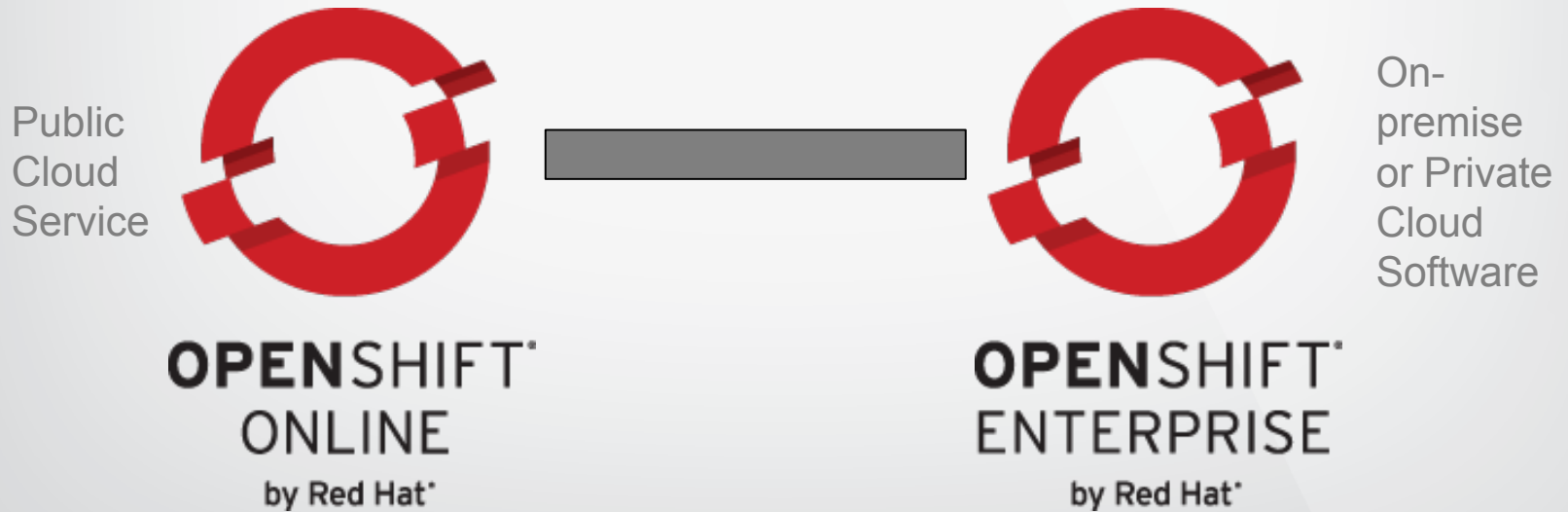
OPENSIFT STRATEGY

Public
Cloud
Service

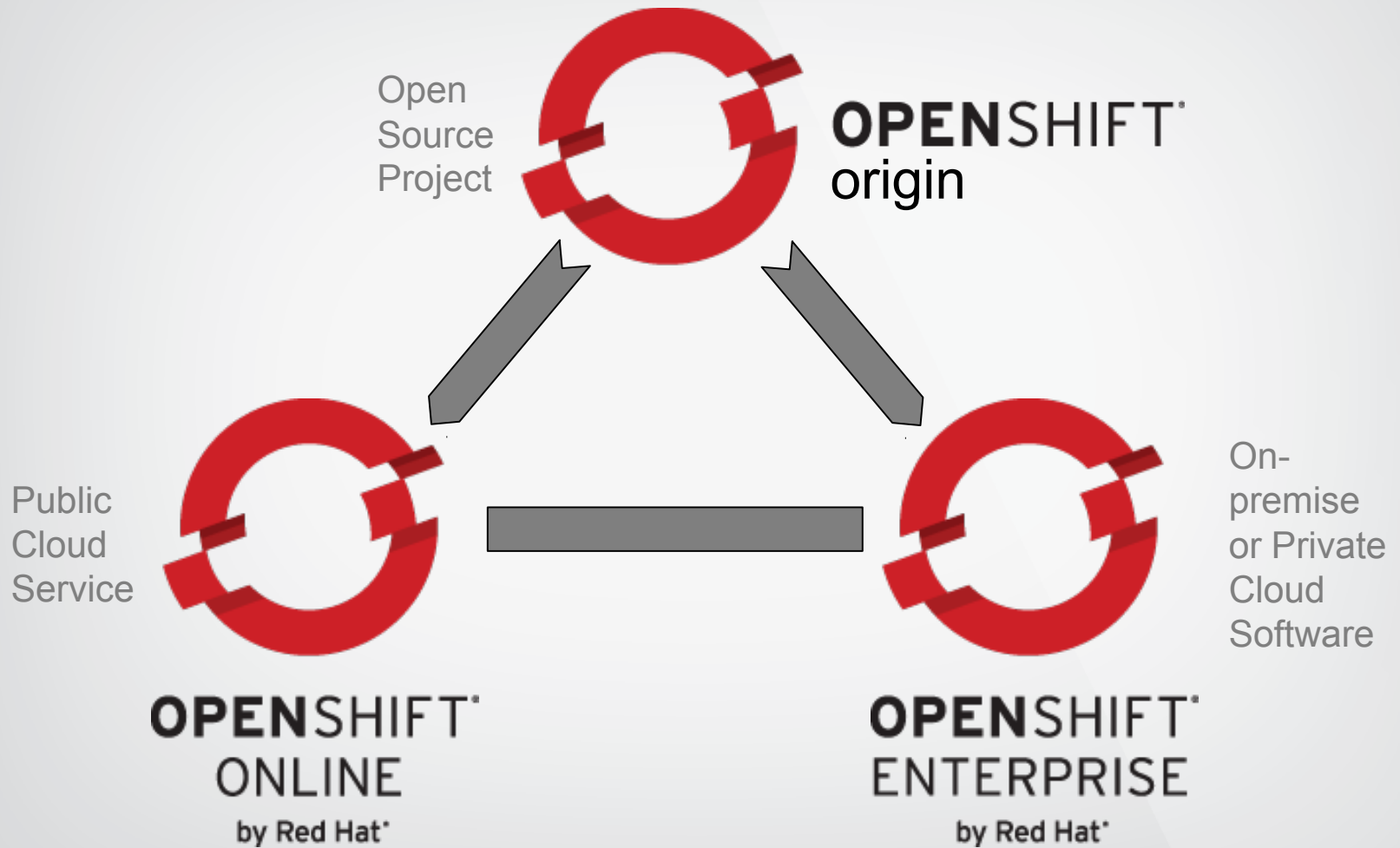


OPENSIFT[®]
ONLINE
by Red Hat[®]

OPENSIFT STRATEGY



OPENSIFT STRATEGY



WEB CONSOLE

Web Cartridges

The web cartridge is the heart of your application, handling incoming web requests and dishing out web pages, business APIs, or the content for your next hot mobile app.

JBoss Enterprise Application Platform 6.0 RECENTLY ADDED

Market-leading open source enterprise platform for next-generation, highly transactional enterprise Java applications. Build and deploy enterprise Java in the cloud.

Select »

Python 2.6

Python is a general-purpose, high-level programming language whose design philosophy emphasizes code readability. Popular development frameworks include: Django, Bottle, Pylons, Zope and TurboGears.

Select »

JBoss Application Server 7.1

The leading open source Java EE6 application server for enterprise Java applications. Popular development frameworks include Seam, CDI, Weld, and Spring.

Select »

Ruby 1.9.3

Ruby is a dynamic, reflective, general-purpose object-oriented programming language. Popular development frameworks include Ruby on Rails and Sinatra.

Select »

Node.js 0.6

Node.js is a platform built on Chrome's JavaScript runtime for easily building fast, scalable network applications. Node.js is perfect for

Do-It-Yourself

The Do-It-Yourself (DIY) application type is a blank slate for trying unsupported languages, frameworks, and middleware on OpenShift. See

CLI? OF COURSE

Create Apps

```
rhc app create -a javasample -t jbossas-7
```

Add MongoDB

```
rhc app cartridge add -a javasample -c mongodb-2.0
```

Add add EAR file to your deployments directory

```
cd javasample
```

```
cp /path/to/ear/earfilename.ear ./deployments
```

Add the EAR file to git

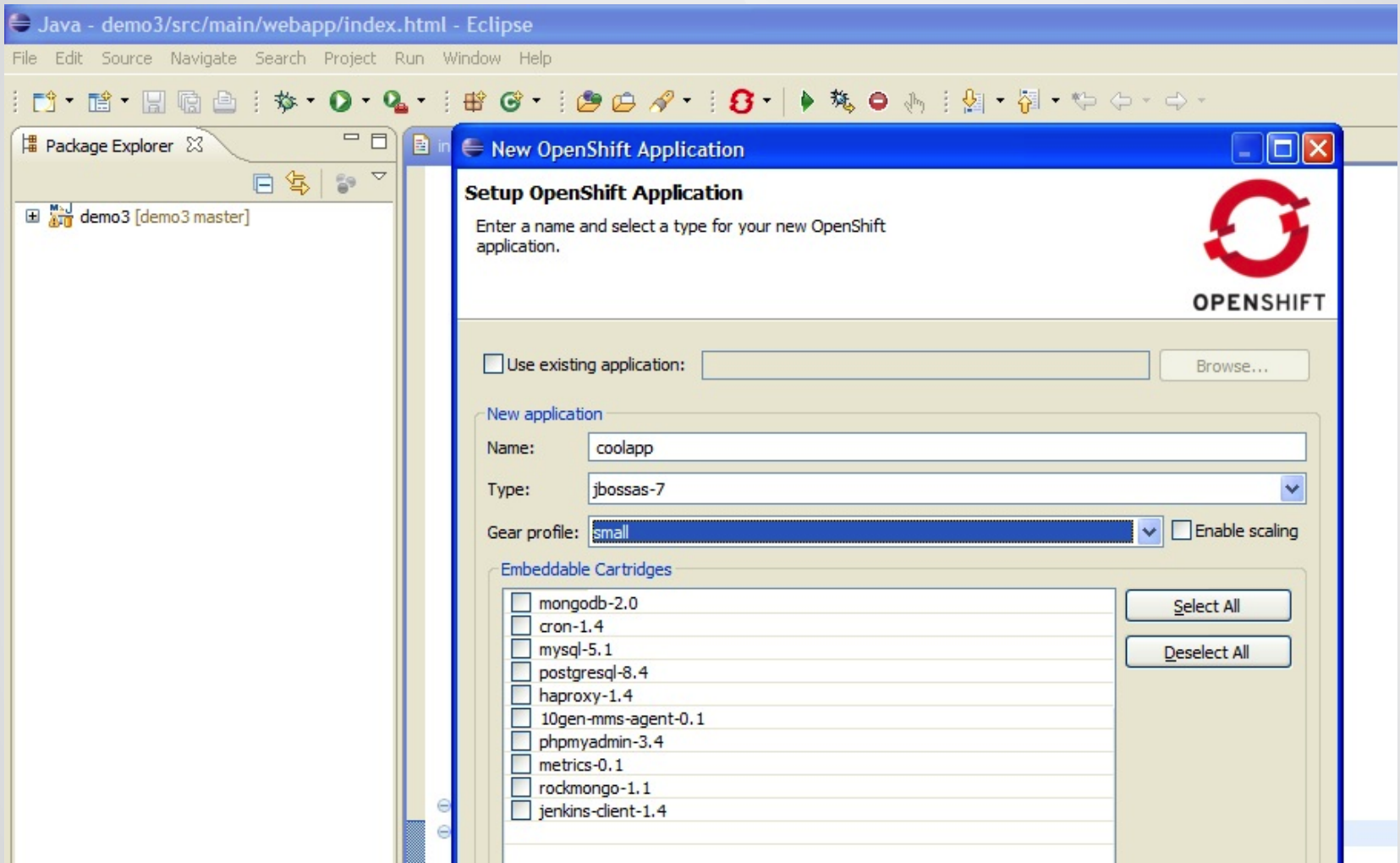
```
git add ./deployments/earfilename.ear
```

Push your code

```
git push
```

Done

ECLIPSE, TOO.



HOW IT WORKS



YES, WE STILL HAVE INFRASTRUCTURE



AWS / CloudForms / OpenStack (IaaS) / RHEV (Virt) / Bare Metal

RHEL IS THE FOUNDATION



OpenShift is Built on Instances of
Red Hat Enterprise Linux (RHEL)

RHEL

RHEL

RHEL

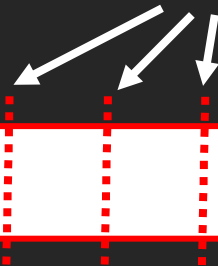
RHEL

AWS / CloudForms / OpenStack (IaaS) / RHEV (Virt) / Bare Metal

RHEL GIVES US MULTI-TENANCY



SELinux Policies securely subdivide
the Node instances.



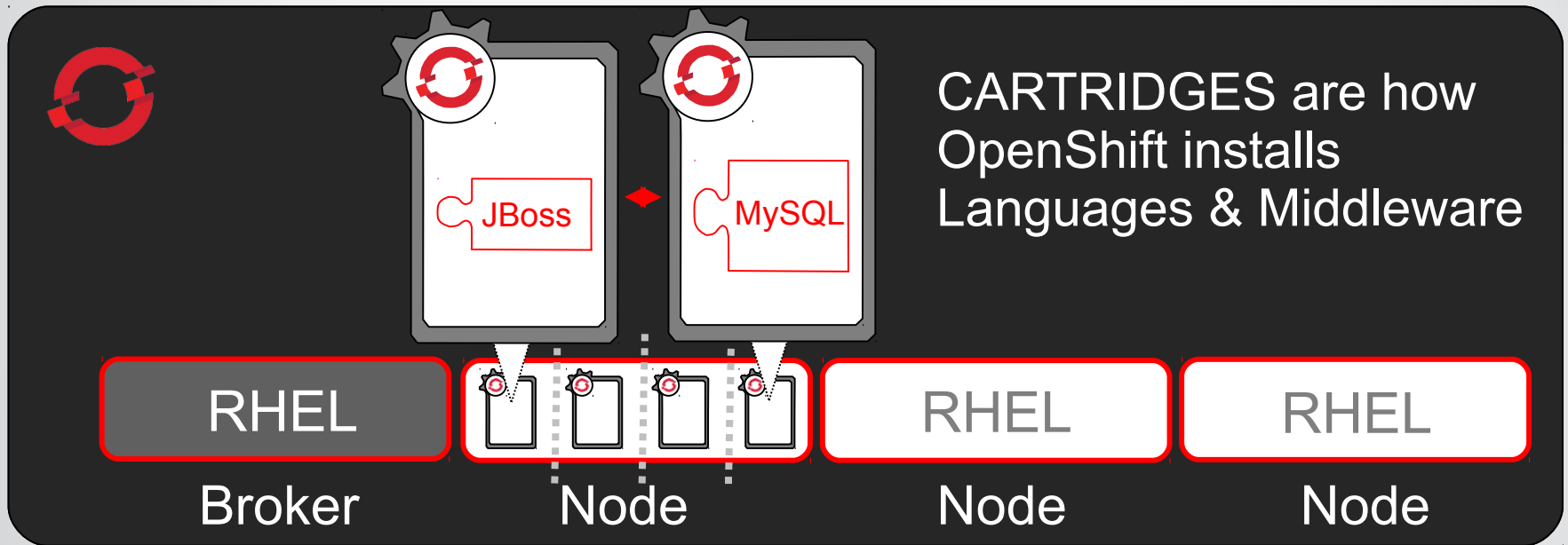
RHEL

RHEL

RHEL

AWS / CloudForms / OpenStack (IaaS) / RHEV (Virt) / Bare Metal

RHEL GIVES US MULTI-TENANCY



AWS / CloudForms / OpenStack (IaaS) / RHEV (Virt) / Bare Metal

CARTRIDGES

Web Cartridges

The web cartridge is the heart of your application, handling incoming web requests and dishing out web pages, business APIs, or the content for your next hot mobile app.

JBoss Enterprise Application Platform 6.0 RECENTLY ADDED

Market-leading open source enterprise platform for next-generation, highly transactional enterprise Java applications. Build and deploy enterprise Java in the cloud.

Select »

Python 2.6

Python is a general-purpose, high-level programming language whose design philosophy emphasizes code readability. Popular development frameworks include: Django, Bottle, Pylons, Zope and TurboGears.

Select »

JBoss Application Server 7.1

The leading open source Java EE6 application server for enterprise Java applications. Popular development frameworks include Seam, CDI, Weld, and Spring.

Select »

Ruby 1.9.3

Ruby is a dynamic, reflective, general-purpose object-oriented programming language. Popular development frameworks include Ruby on Rails and Sinatra.

Select »

Node.js 0.6

Node.js is a platform built on Chrome's JavaScript runtime for easily building fast, scalable network applications. Node.js is perfect for

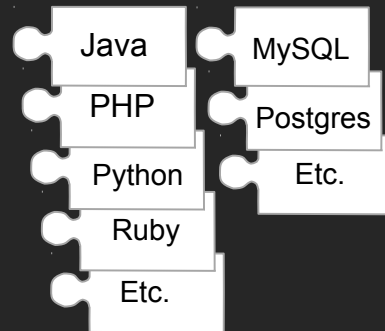
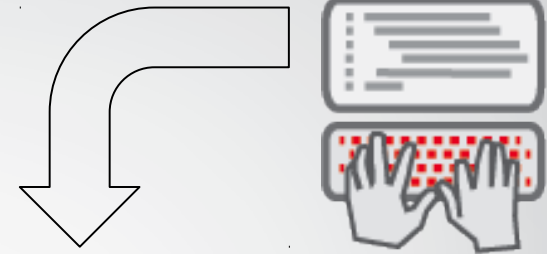
Do-It-Yourself

The Do-It-Yourself (DIY) application type is a blank slate for trying unsupported languages, frameworks, and middleware on OpenShift. See

EVERYTHING DEVELOPERS ALREADY USE



YES, YOU CAN BUILD YOUR OWN



OpenShift Default
Cartridges



Developers can add custom
language, data-store, or
middleware with with a custom
Cartridge.

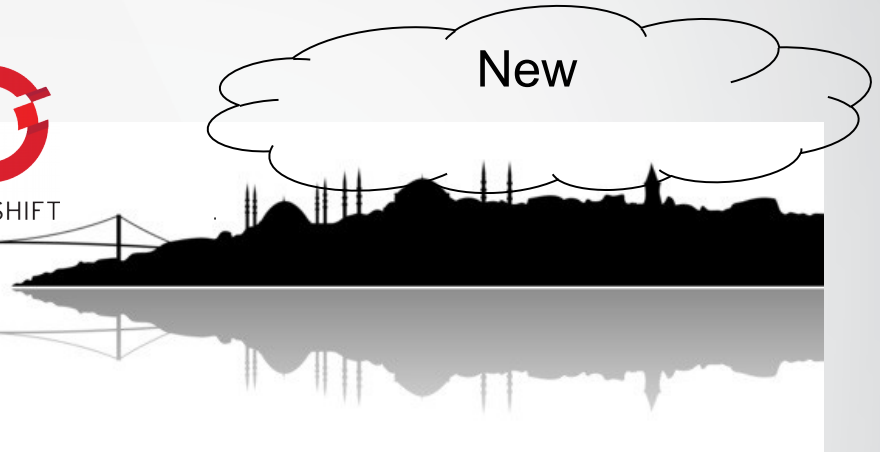
AWS / CloudForms / OpenStack (IaaS) / RHEV (Virt) / Bare Metal

Established



OPENSIFT

New



**I DON'T ALWAYS TEST MY
CODE**

**BUT WHEN I DO,
IT'S DURING LIVE
DEMOS**

memegenerator.net