Cloud Foundry and Spring Data for MongoDB Java developers

CLOUD FOUNDRY

Thomas Risberg, Cloud Foundry Team, trisberg@vmware.com

October, 2011

vmware

Why am I here? Spring!

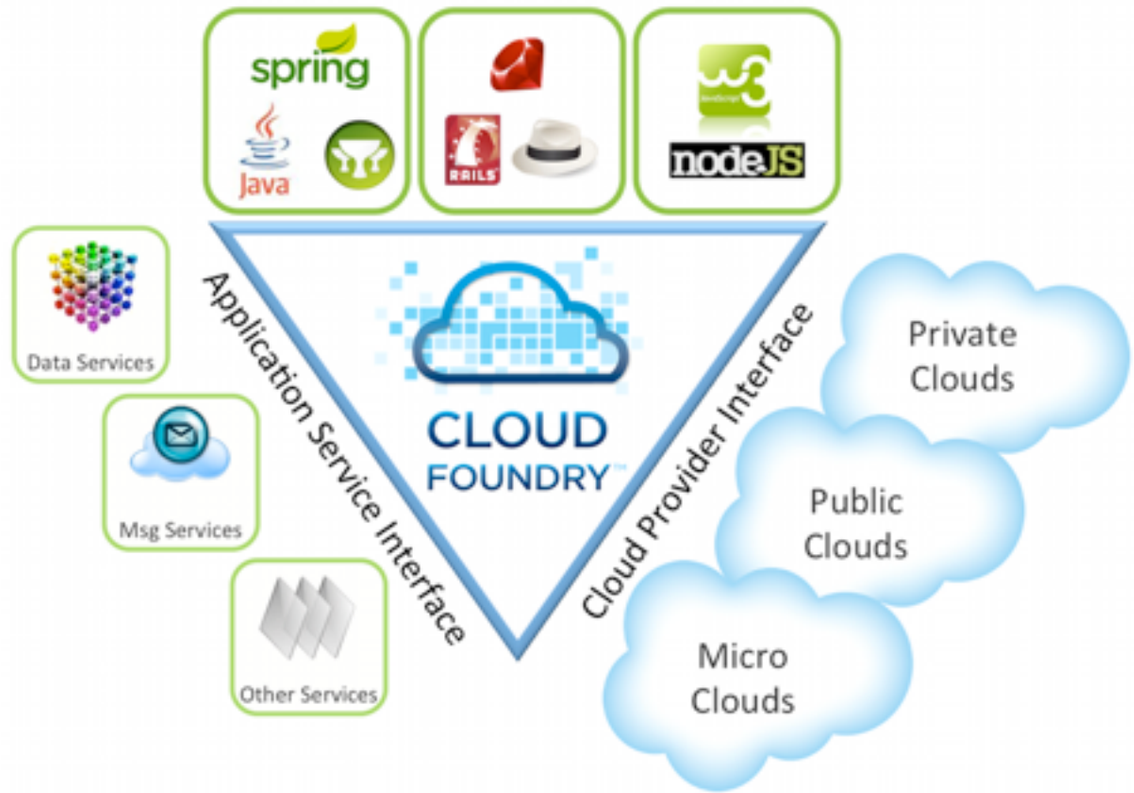
- I've been involved with the Spring Framework since 2003
  - A framework that was providing a light-weight Java alternative to the heavy-weight app-servers that dominated the enterprise Java space
  - Simplified developer experience
  - Application developer driven and community oriented
  - We were successful and today many large enterprises deploy large mission critical applications on Tomcat using Spring
  - Core values:
    - Portability
    - Productivity
    - Simplicity

Why am I here? MongoDB!

- MongoDB and other NOSQL databases have many similarities
  - Provide a lighter weight alternative that can perform and scale better
  - Trying to push into the enterprise application space
  - Going up against established products promoted by Microsoft, IBM and Oracle

Why am I here? Cloud Foundry!

- We are also seeing a shift away from companies running their own servers, instead relying on services available in the “cloud”
  - Simplified “middleware” stack
  - Don’t waste time installing app servers and database servers
- This makes for a disruptive “perfect storm” scenario
  - Apps persisting data to NOSQL databases
  - All running in the cloud
  - You worry about your business logic, someone else worries about the infrastructure and configuration etc.
  - Developer centric
  - Community driven

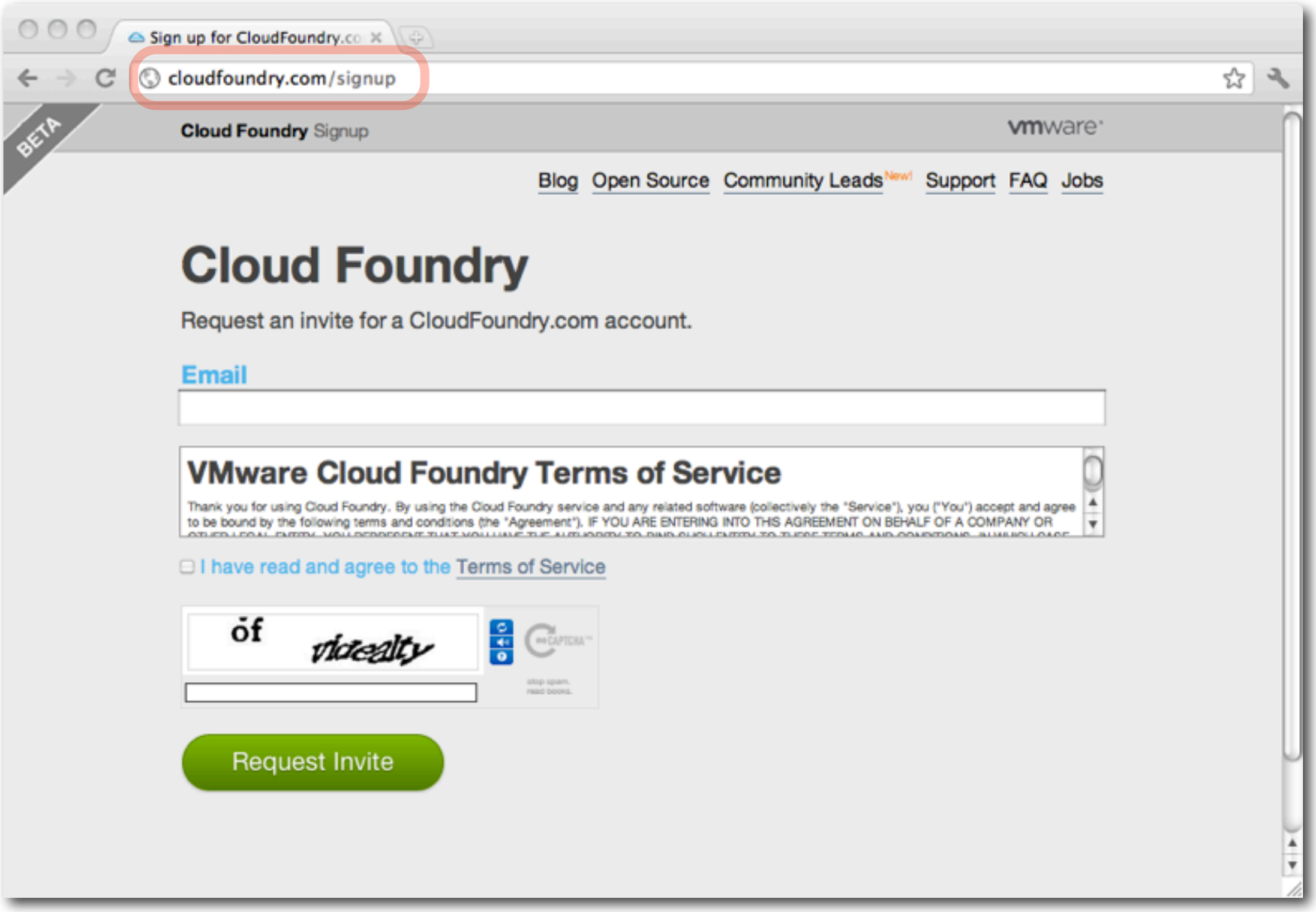


vmware

What is Cloud Foundry?

- Cloud Foundry is a PaaS
  - The application platform will be delivered as a service in the cloud era
  - The industry calls this platform as a service (PaaS)
- PaaS makes it much easier to deploy, run and scale applications
- But PaaS solutions in the market have fatal flaws today
  - Limited in framework, application services and/or cloud support
- Cloud Foundry aim to fix that…

Cloud Foundry Installation - Signup



Cloud Foundry Installation - vmc client

Terminal — bash — 80×24

```
Bermuda:~ trisberg$ sudo gem install vmc --no-rdoc --no-ri
Successfully installed vmc-0.3.12
1 gem installed
Bermuda:~ trisberg$ vmc target api.cloudfoundry.com
Succesfully targeted to [http://api.cloudfoundry.com]

Bermuda:~ trisberg$ vmc login --email trisberg@vmware.com
Password: ********
Successfully logged into [http://api.cloudfoundry.com]

Bermuda:~ trisberg$ vmc info

VMware's Cloud Application Platform
For support visit http://support.cloudfoundry.com

Target:   http://api.cloudfoundry.com (v0.999)
Client:   v0.3.12

User:     trisberg@vmware.com
Usage:    Memory   (1.9G of 2.0G total)
          Services (13 of 16 total)
          Apps     (5 of 20 total)

Bermuda:~ trisberg$
```

Cloud Foundry - your first MongoDB service

Terminal — bash — 80×25

```
Bermuda:mongo-java-web trisberg$ vmc create-service mongodb my-mongo
Creating Service: OK

Bermuda:mongo-java-web trisberg$ vmc push mongo-java-web --path target
Application Deployed URL: 'mongo-java-web.cloudfoundry.com'?
Detected a Java Web Application, is this correct? [Yn]:
Memory Reservation [Default:512M] (64M, 128M or 256M) 256M
Creating Application: OK
Would you like to bind any services to 'mongo-java-web'? [yN]:
Uploading Application:
  Checking for available resources: OK
  Processing resources: OK
  Packing application: OK
  Uploading (2K): OK
Push Status: OK
Staging Application: OK
Starting Application: OK

Bermuda:mongo-java-web trisberg$ vmc bind-service my-mongo mongo-java-web
Binding Service: OK
Stopping Application: OK
Staging Application: OK
Starting Application: OK

Bermuda:mongo-java-web trisberg$
```


vmware

Cloud Foundry Demo

LIVE DEMO

https://github.com/trisberg/mongoboston-cloudfoundry

vmware

Characteristics of PaaS

The application platform for the cloud era

- Integrated software stack
- Application execution engine
- Self-service application deployment
- Automated application infrastructure provisioning
- Curated, updated and operated as a service

Cloud Foundry – The first open PaaS

Self-service application execution engine

- Build applications with latest high productivity frameworks

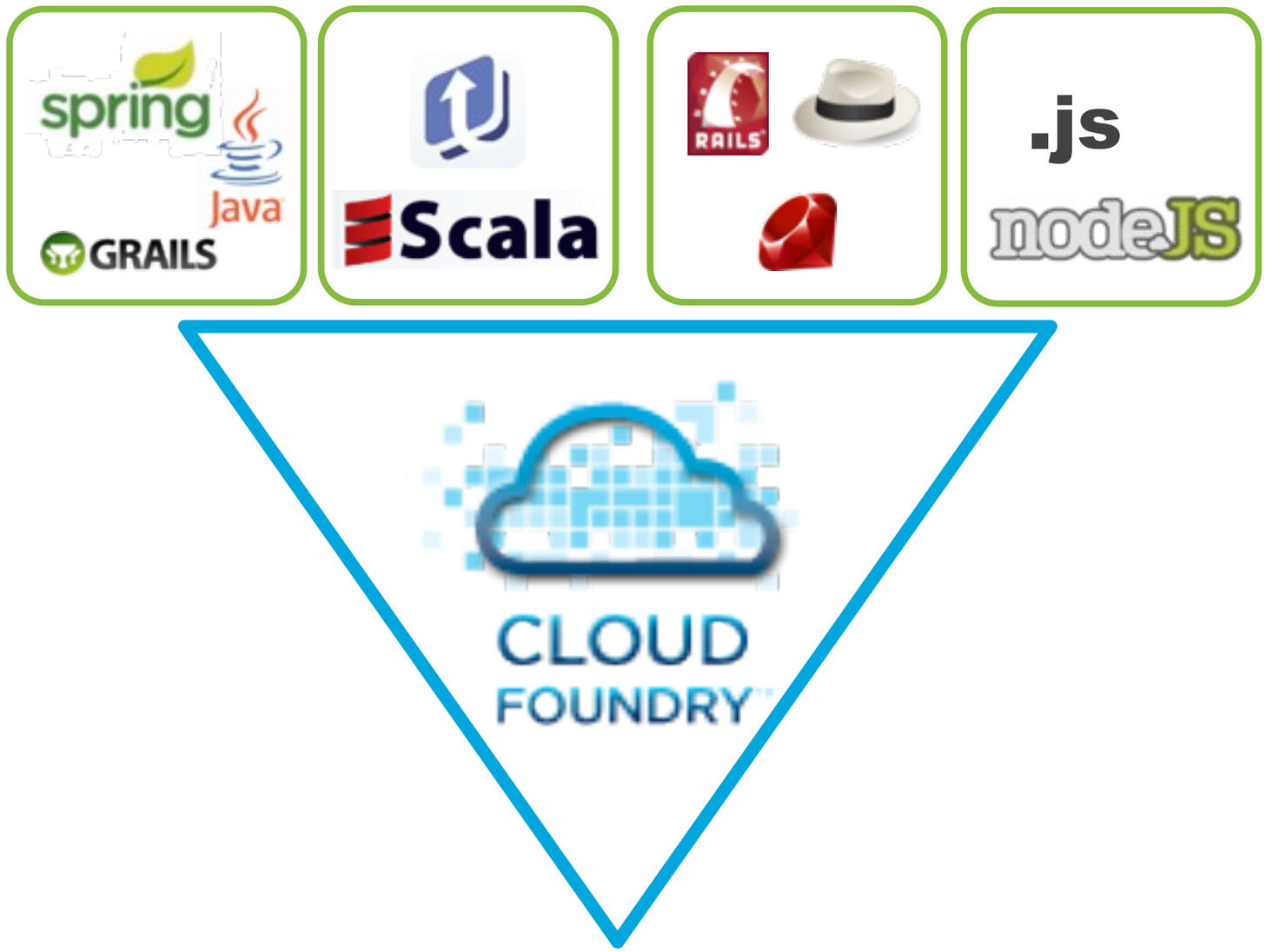
Automation engine for deployment and lifecycle management

- Deploy and cloud-scale applications in seconds

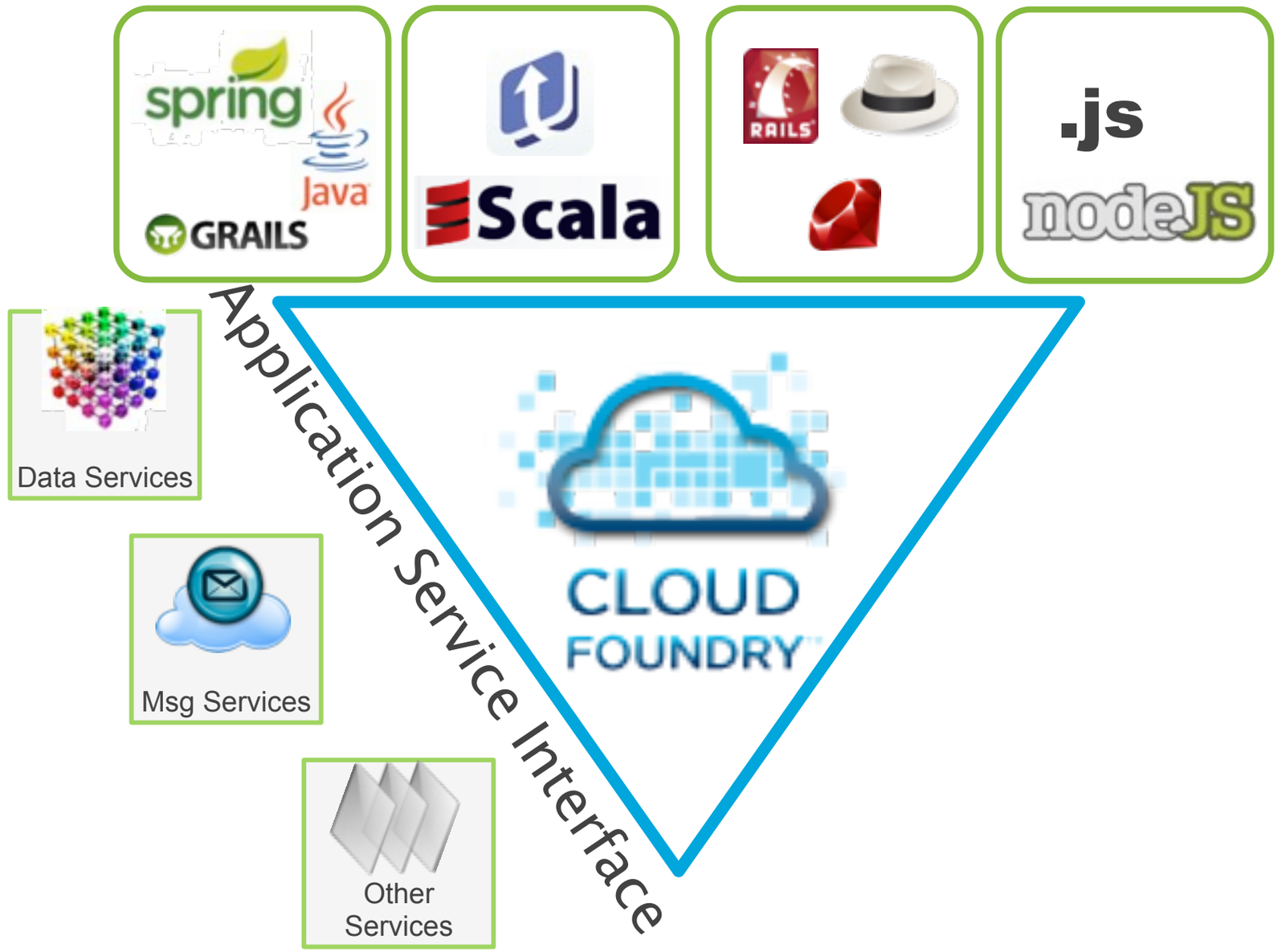
Open architecture

- Choice of clouds for deployment
- Choice of industry-standard frameworks
- Choice of application infrastructure services
- Extensible architecture to “digest” future cloud innovation
- Available as open source

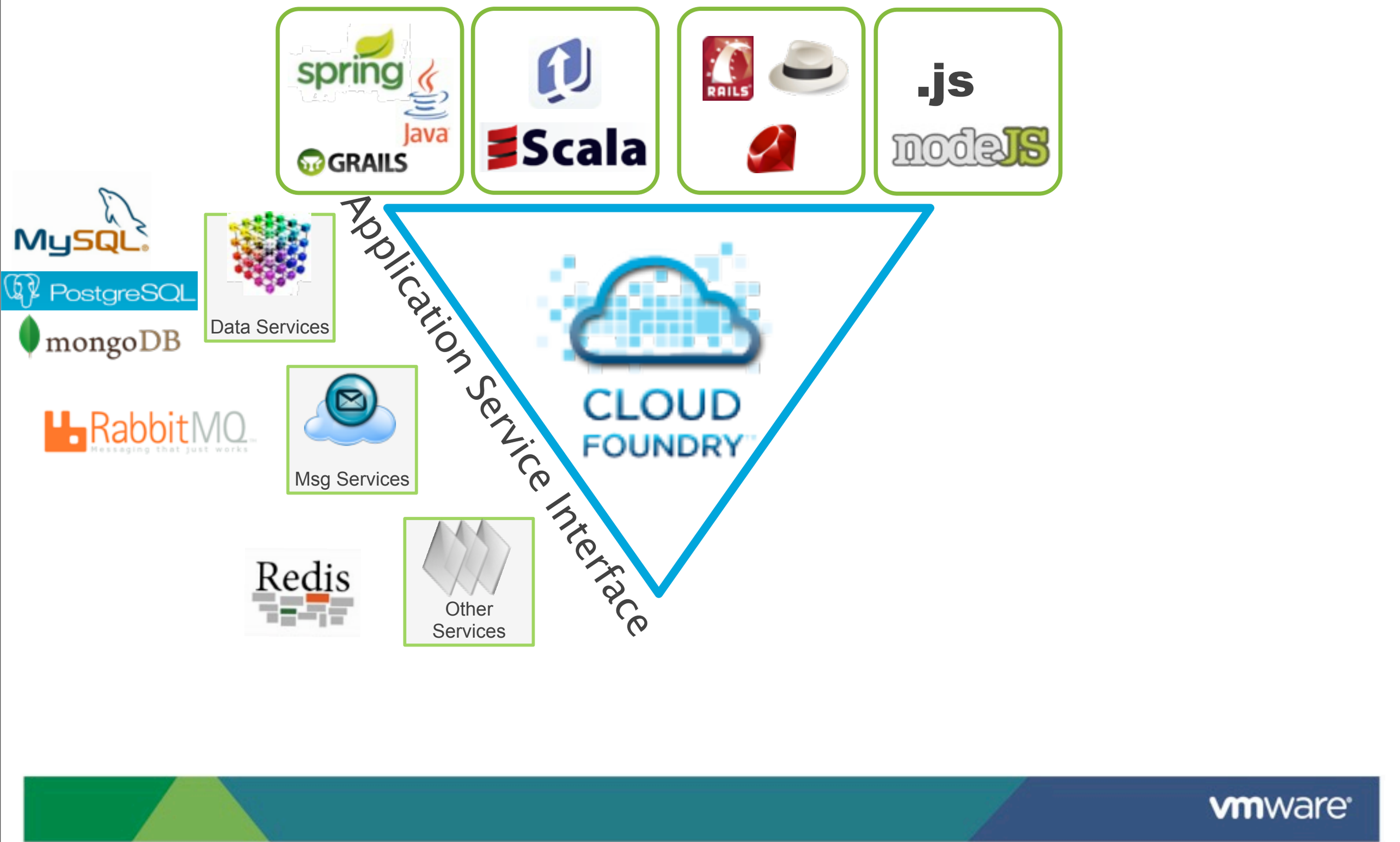
Choice of frameworks



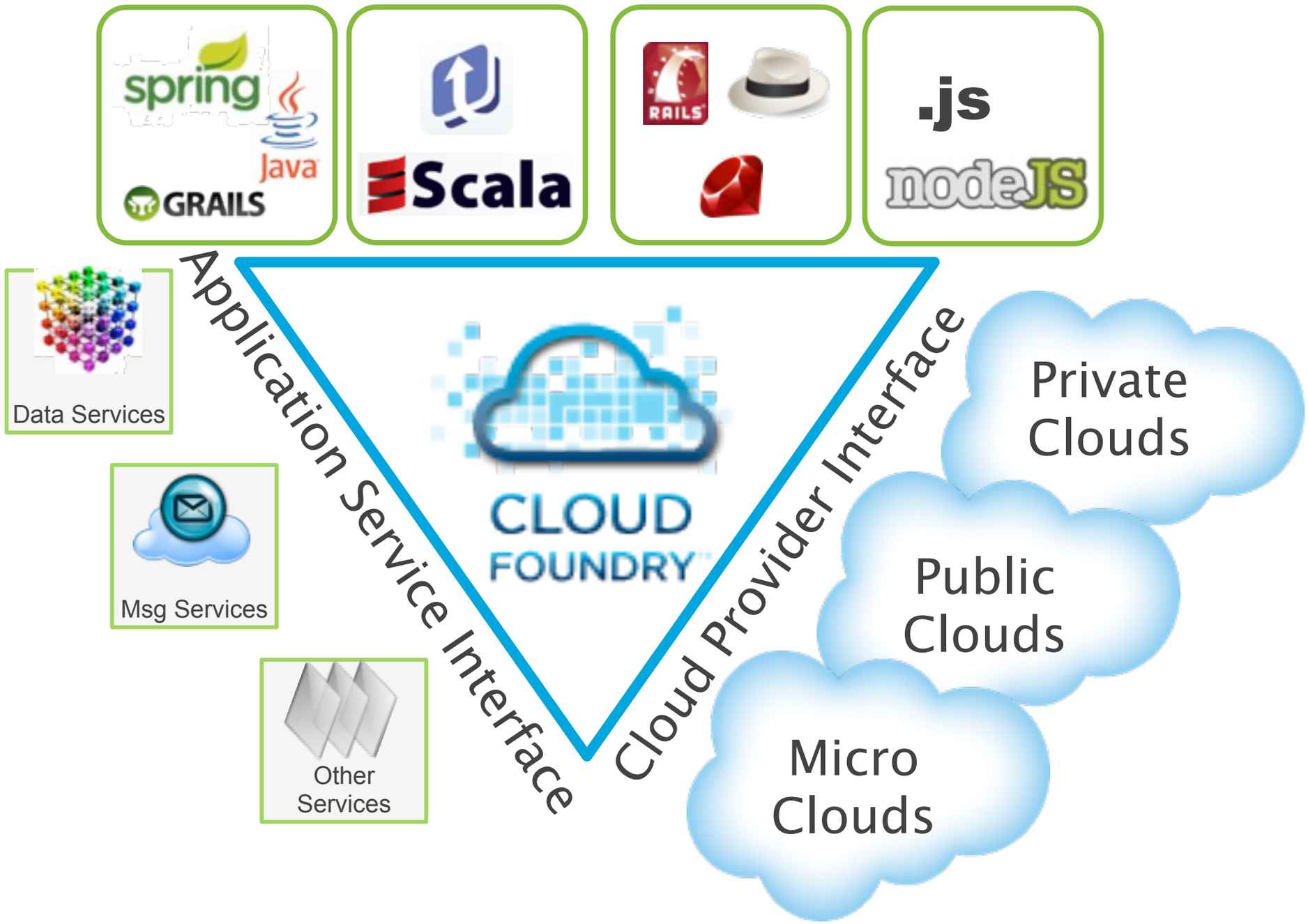
Choice of application services



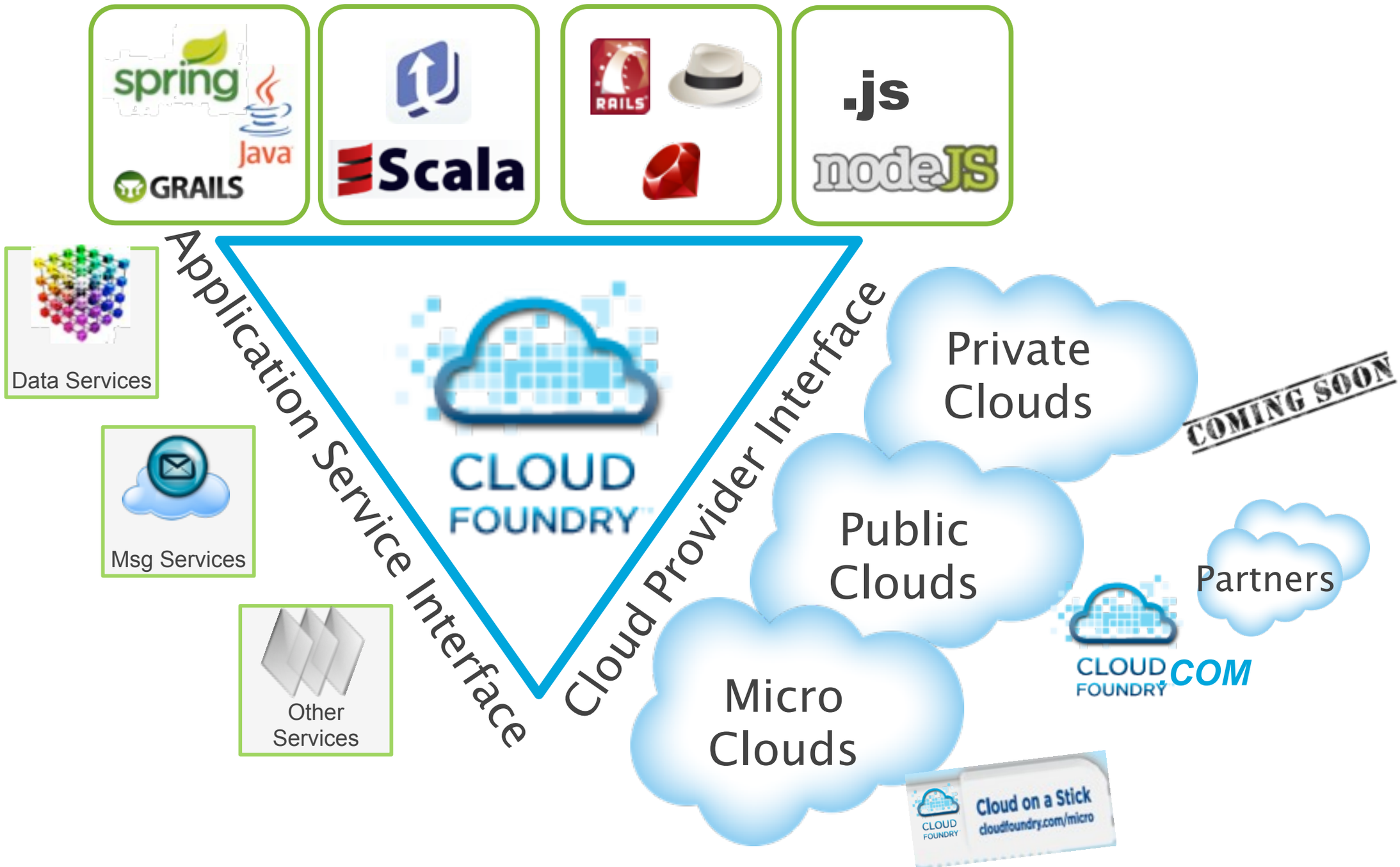
Choice of application services



Choice of clouds



Choice of clouds



Cloud Foundry: Four key initiatives

CloudFoundry.org
Open Source Project

CloudFoundry.com
Public Cloud Service

CLOUD FOUNDRY™

“Project Bento”
Enterprise PaaS on vSphere

COMING SOON

Micro Cloud Foundry™
Developer Download

vmware

CloudFoundry.com (beta)

- Multi-tenant PaaS service for grass-root developers
  - Operated by VMware
  - Multiple development frameworks
    - Spring, Ruby (Rails, Sinatra) Node.js, Scala, Grails
  - Multiple application services
    - vFabric RabbitMQ, vFabric Postgres, MySQL, Redis, MongoDB
  - Multiple developers tools
    - Command line interface (‘vmc’) , Eclipsed based IDE (‘sts’)

- Free signup at www.cloudfoundry.com
  - Paid GA offering to be available in FY12

Micro Cloud Foundry™ (beta)

Industry-first downloadable PaaS

- Instance of Cloud Foundry that runs on a developer's PC
- Distributed as a virtual machine image
  - Vmware Fusion (Mac OS X) , VMware Player (Linux, Windows, free), VMware Workstation (Windows)
- Symmetrical with other Cloud Foundry instances
  - Develop locally, deploy to the cloud
- Dynamic DNS support
  - Run Micro Cloud Foundry at home, office or coffee shop – without any reconfiguration
- Updated regularly

- Downloadable from micro.cloudfoundry.com

"Honey I shrank the cloud"

CloudFoundry.org – it's open source!

- Community open-source project on www.cloudfoundry.org
  - Expectation for modern developer technologies
  - Source code on GitHub
  - Apache 2 license
- Allows any developer to
  - Access, evaluate and modify the code
  - Integrate other frameworks
  - Add application services
  - Deploy to other infrastructure clouds
- Governance follows the Spring open source model

Cloud Foundry “Project Bento”

- Enterprise version of Cloud Foundry for private clouds
  - Lets IT run as a service (or have hosted on their behalf)
  - Runs on VMware virtual infrastructure
  - Includes vFabric application services
  - Backed by enterprise-class support
  - Hybrid cloud symmetry
  - Service Broker provides gateway to existing enterprise systems

- To be available in FY12

Significant adoption and a growing ecosystem

Significant adoption and a growing ecosystem

- **CloudFoundry.com**
  - Tens of thousands of beta users
  - Thousands of applications deployed

Significant adoption and a growing ecosystem

- **CloudFoundry.com**
  - Tens of thousands of beta users
  - Thousands of applications deployed

- **Strong community participation**
  - Hundreds of contributions from the OSS community
  - Erlang, install scripts, Neo4J, Rack, JRuby, PHP, Python

Significant adoption and a growing ecosystem

- **CloudFoundry.com**
  - Tens of thousands of beta users
  - Thousands of applications deployed

- **Strong community participation**
  - Hundreds of contributions from the OSS community
  - Erlang, install scripts, Neo4J, Rack, JRuby, PHP, Python

- **Multiple distributors, deployers, clouds**
  - Canonical, Dell, enStratus, OpsCode, RightScale

Significant adoption and a growing ecosystem

- **CloudFoundry.com**
  - Tens of thousands of beta users
  - Thousands of applications deployed

- **Strong community participation**
  - Hundreds of contributions from the OSS community
  - Erlang, install scripts, Neo4J, Rack, JRuby, PHP, Python

- **Multiple distributors, deployers, clouds**
  - Canonical, Dell, enStratus, OpsCode, RightScale

- **Cloud Foundry Community Leads Program**
  - Charter members: PHP by PHPfog, Python by ActiveState

Spring Framework

The Java application framework for the cloud era

- Bean Factory / Application Context support
- Configuration in XML or Java code with annotations
  - Spring 3.1 introduces profiles
    - easy way to use different configurations in a variety of environments
- Strong Data Access support
- Web MVC framework
- and much more ...

- Project Home: http://www.springsource.org/about

Spring Projects

Large “ecosystem” of related Spring Projects:

- Web Flow
- Spring Security
- Spring Web Services
- Spring Integration
- Spring Batch
- Spring Data
- ...

Spring Framework - Data Access

Built-in data access support:

- Transaction abstractions, Data access exceptions
- JDBC - JdbcTemplate
- ORM - Hibernate, JPA support
- OXM - Object to XML mapping
- Cache support (Spring 3.1)

Spring Data Umbrella Project

Long and growing list of sub-projects:

- **Spring Data Neo4j (Graph)**
- **Spring Data Redis**
- **Spring Data MongoDB**
- Spring Data Column
- Spring Data Blob
- **Spring Data JPA Repository**
- **Spring Data JDBC Extensions**
- **Spring Gemfire**
- **Spring Hadoop ...**
- **Grails NOSQL support (grails-data-mapping)**

What is Spring Data

What is Spring Data all about?

- Bring classic Spring value propositions to NOSQL
  - ✓Productivity
  - ✓Programming model consistency
- Support for a wide range of NOSQL databases
- Also, support for new features for JDBC and JPA
- Support for other data related projects like Hadoop and Gemfire

Spring Data MongoDB

Spring Data support for MongoDB:

- MongoTemplate
  - ✓ MongoConverter interface for mapping Mongo documents
    - Built-in Advanced Mapping
      - Annotation based (@Document, @Id, @DbRef)
    - MappingMongoConverter for POJO mapping support
    - Leverage Spring 3.0 TypeConverters and SpEL
  - ✓ Exception translation
  - ✓ Java based Query, Criteria, and Update DSLs
- MongoRepository
  - ✓ Built on Spring Data JPA (Hades) support for JPA Repositories
  - ✓ QueryDSL integration to support type-safe queries.

Spring Data MongoDB releases

Spring Data MongoDB releases:

- Current: 1.0.0.M4
  - this is also the currently supported version for Cloud Foundry
- Release Candidate 1.0.0.RC1
  - planned for Mongo Berlin October 10th
- GA Release 1.0.0.RELEASE
  - planned for SpringOne2GX October 26th

```
<dependency>
     <groupId>org.springframework.data</groupId>
     <artifactId>spring-data-mongodb</artifactId>
     <version>1.0.0.M4</version>
</dependency>
...
<repository>
     <id>org.springframework.maven.milestone</id>
     <name>Spring Maven Milestone Repository</name>
     <url>http://maven.springframework.org/milestone</url>
     <snapshots><enabled>false</enabled></snapshots>
</repository>
```

Finding Spring Data MongoDB

Where can you find more about Spring Data MongoDB?

- GitHub:
  - https://github.com/SpringSource
- Web page:
  - http://www.springsource.org/spring-data/mongdb
- Mailing List
  - https://lists.springsource.com/listmanager/listinfo/spring-data-mongodb
- Forum:
  - http://forum.springsource.org/forumdisplay.php?f=80

vmware

Spring Data and MongoDB Demo

LIVE DEMO

https://github.com/trisberg/mongoboston-mongotemplate

vmware

Spring Data Repository

Spring Data Repository basics:

- Generic repository implementation
- Basic CRUD (create, read, update and delete) methods
- Generating code for queries defined in repository interface
  - findAll
  - findByName ...
- Pagination and sorting support
- Currently has JPA and Mongo implementations

Spring Data Repository

CrudRepository

long	**count**() Returns the number of entities available.
void	**delete**(ID id) Deletes the entity with the given id.
void	**delete**(Iterable<? extends T> entities) Deletes the given entities.
void	**delete**(T entity) Deletes a given entity.
void	**deleteAll**() Deletes all entities managed by the repository.
boolean	**exists**(ID id) Returns whether an entity with the given id exists.
Iterable<T>	**findAll**() Returns all instances of the type.
T	**findOne**(ID id) Retrives an entity by its primary key.
Iterable<T>	**save**(Iterable<? extends T> entities) Saves all given entities.
T	**save**(T entity) Saves a given entity.

PagingAndSortingRepository

Page<T>	**findAll**(Pageable pageable) Returns a Page of entities meeting the paging restriction provided in the Pageable object.
Iterable<T>	**findAll**(Sort sort) Returns all entities sorted by the given options.

Spring Data Repository query methods:

Keyword	Sample	Logical result
GreaterThan	findByAgeGreaterThan(int age)	{"age" : {"$gt" : age}}
LessThan	findByAgeLessThan(int age)	{"age" : {"$lt" : age}}
Between	findByAgeBetween(int from, int to)	{"age" : {"$gt" : from, "$lt" : to}}
IsNotNull, NotNull	findByNameNotNull()	{"name" : {"$ne" : null}}
IsNull, Null	findByNameNull()	{"name" : null}
Like	findByNameLike(String expr)	{"name" : expr} (expr as regex)
(No keyword)	findByName(String name)	{"name" : name}
Not	findByNameNot(String name)	{"name" : {"$ne" : name}}

Spring Data Mongo Repository Demo

LIVE DEMO

https://github.com/trisberg/mongoboston-repository

vmware

Questions?