Federal Geographic Data Committee
GeoCloud Platform

Presentation of Business Use Case

Douglas Nebert, US Geological Survey
Robert Patt-Corner, eGlobalTech/GSA

4/7/11
Agenda
Cloud Community Platform Operation

- Business Overview
  - Leveraging Infrastructure Benefits for Geographical Applications
  - GeoCloud Goals and Activities
  - Primary Activity Cycle
  - Current Platform Architecture

- Use Case Walkthrough
  - Overview: Goals and Roles
  - Issues in Platform Deployment
  - Use Case 1: Create, Compose and Harden Target Platforms
  - Use Case 2: Maintain and Monitor Target Platforms
  - Use Case 3: Develop Project Application Deployment Packages
Introducing the GeoCloud Community Platform
Leveraging Infrastructure Benefits for Geographical Applications

Platform Savings
- Platform building time & effort
- Reduced maintenance costs
- Faster deployment
- Cost effective development / test

Infrastructure as A Service Savings
- Hardware Savings
- Operations Savings
- Scalability Savings

Large pool of agency applications identified for cloud migration

GeoCloudCommunity Platform

Apps Need Platforms

Platform Reference Implementations

Platforms Amplify IaaS Savings

More...
## Deploying Federal Geospatial Services in the Cloud

### GeoCloud Goals and Activities

One-year project to test and monitor externally-hosted Cloud data and service solutions for the geospatial domain, sponsored through the Geospatial Line of Business

<table>
<thead>
<tr>
<th>Architectures</th>
<th>Acquire, compose and deploy reference platform prototypes in commercial and private cloud</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Identify requirements-driven solution architectures and platforms for various sized deployments of geospatial data and services</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Cost Models</th>
<th>Monitor costs, loads, issues and options</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Document and assess cost models to support scalability, reliability, and redundancy</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Certification</th>
<th>Certify Geospatial Solution Packages to facilitate reuse</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Expedite FISMA certification and accreditation for agency adoption of packaged solution architectures</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Comparisons</th>
<th>Document Best Practices and guides to agencies on adoption of geospatial Cloud infrastructure</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Support and collect cost comparison information from agencies for existing and externally-hosted Cloud solutions</td>
<td></td>
</tr>
<tr>
<td>• Document lessons learned and best practices</td>
<td></td>
</tr>
</tbody>
</table>
Cloud Community Platform Service Activities
Primary Activity Cycle

Create Prototype Platforms

Prototype
- Create series of Federal Cloud Computing reference platform prototypes
- Support the wide range of target apps

Validate
- Ensure platforms work with identified agency applications

Document and Promulgate

Document
- Document best practices and lessons learned
- Document cost models and benchmarks
- Bundle the platforms and artifacts
- Promulgate to Federal Cloud Community of Interest

Validate With Agency Apps

Create Prototypes Platforms
Cloud Community Platform Service Activities
Platform Architecture

Basic Image
- Windows Server Image .NET, IIS

Windows Platforms
- Base Linux 64 (of known provenance)

Open Source Platforms
- Base Linux 64 (of known provenance)
- Open Source Core
  - Apache, PostGreS/MySQL, Java, Ruby on Rails, Tomcat, Perl, Python

Base Platforms
- Open Source Additions
  - Java, PHP, PostGRES
  - Tomcat, FLEX, Apache

Specialized Platforms
- ESRI ArcGIS
  - Geospatial Platform
- Additional Open Source
  - GlassFish 3, Axis, SunMQ
- Additional Geospatial
  - THREDDS, More

Tiers (e.g. database, app server) can be split or combined as needed
Cloud Community Platform Service Scope
Initial Target Applications for Community Platforms

<table>
<thead>
<tr>
<th>Agency</th>
<th>Project</th>
<th>Target Platform</th>
<th>Status</th>
<th>Base Platform</th>
</tr>
</thead>
<tbody>
<tr>
<td>DOI</td>
<td>Wetlands Mapper</td>
<td>Windows/IIS/ESRI</td>
<td>Pending</td>
<td>Windows ESRI Platform with additions for ORACLE as needed</td>
</tr>
<tr>
<td>USDA</td>
<td>USDA CDSI Web Services</td>
<td>Windows/IIS/ESRI/SQLServer</td>
<td>&quot;</td>
<td>&quot;</td>
</tr>
<tr>
<td>USDA</td>
<td>USDA Geospatial Public Catalog</td>
<td>Windows/JSP/ESRI/SQL2008</td>
<td>&quot;</td>
<td>&quot;</td>
</tr>
<tr>
<td>EPA</td>
<td>EPA Lakes &amp; Ponds</td>
<td>Windows/PostGres/ESRI</td>
<td>&quot;</td>
<td>&quot;</td>
</tr>
<tr>
<td>NOAA</td>
<td>NOAA Particles</td>
<td>Linux/Java/Axis/MQ</td>
<td>Sandbox</td>
<td>Linux Open Source Platform with Java/Ruby on Rails/Tomcat,/EJB3, PostGres, MySQL (alt) Python, PHP, Perl and Apache Good candidate for a single base platform</td>
</tr>
<tr>
<td>USGS</td>
<td>GEOSS GeoNetwork w. additions</td>
<td>Linux/TomCat/PostGres</td>
<td>Sandbox</td>
<td>&quot;</td>
</tr>
<tr>
<td>Census</td>
<td>TIGER/LINE Shapefiles</td>
<td>Linux / Apache / Perl/Python</td>
<td>Graduating</td>
<td>&quot;</td>
</tr>
<tr>
<td>NOAA</td>
<td>IOOS Registry +</td>
<td>Linux/TomCat/PostGres</td>
<td>Pending</td>
<td>&quot;</td>
</tr>
<tr>
<td>NOAA</td>
<td>NOAA ERDAP</td>
<td>Linux/TomCat/THREDDS</td>
<td>Pending</td>
<td>&quot;</td>
</tr>
<tr>
<td>FGDC</td>
<td>Community Portal</td>
<td>Linux LAMP</td>
<td>Production</td>
<td>&quot;</td>
</tr>
<tr>
<td>USGS</td>
<td>USGS VGI OpenStreetMap</td>
<td>Linux/Apache/PostGIS/Ruby On Rails/Timecop</td>
<td>Pending</td>
<td>&quot;</td>
</tr>
</tbody>
</table>
Overview: FGDC Platform Goals and Roles

- Shared platform(s) providing consumers with ability to directly deploy and operate per NIST model

- Two Principal Roles in Initiative
  - Platform maintainers act as platform providers
  - Platform consumers act as platform users

- Maintainers responsible for platform and its updates

- Consumers responsible for deployment and applications

- Joint responsibility for platform enhancements
Issues in Platform Deployment

Construction and Maintenance
- Designate and/or harden infrastructure
- Construct Platforms
- Maintain common platforms

Orchestration
- Declare the parts of the application and their relationship to each other
- Define the permissions each part has with respect to the other parts
- Provision the parts with permissions as a unified whole

Bootstrapping
- Acquire and access “this time only” startup information
- Configure environment for an application tier
- Use startup information to configure and start the tier

Configuration
- Automatically configure all resources based on startup information
- Start all application tier services
Use Case 1: Create, Compose and Harden Target Platforms

- **Actors:** Platform Providers (Platform Managers and Developers)

- **Goal:** Provide a working initial Geospatial Platform release

- **Brief Description:** A noncloud operating system is hardened to security specifications

- The operating system is ported to the cloud environment as a private FGDC offering

- Requisite platform enablers are installed, configured, scripted, hardened and saved as a virtual machine image to form a point release of a flavor of the Geospatial platform

- Platform developers document lessons learned, best practices and installation operations as part of the deliverables of the project

- The platform release is advertised to consumers of the GeoSpatial Platform through the project portal
Use Case 2: Maintain and Monitor Target Platforms

- **Actors:** Platform Providers (Platform Managers and Developers)

- **Brief Description:** Relevant information sources are monitored for security update and platform enabler enhancement information.

- Platform clients are engaged in a regular process to solicit desired changes and enhancements to the platform based on new project needs, new projects added or current deficiencies.

- On a regular basis, or as an emergency activity in cases of high risk events, the platform is updated to reflect security and/or functionality enhancements.

- Release notes and lessons learned for each point release are documented.

- Changes are advertised to Platform Consumers, with a time window for migration.
Use Case 3: Develop Project Application Deployment Packages

- **Actors:** Platform Consumers

- **Brief Description:** Application deployers access a version of the appropriate Geospatial Platform and install their application on the platform, noting all activities, dependencies and issues.

- Application deployers inform Project Developers of any platform deficiencies through the project portal.

- Application deployers analyze the install activities and create configuration scripts and deployment orchestration templates.

- Deployers organize application, code and template into deployment package.

- Application deployers iteratively test, benchmark and refine the deployment packages to an acceptable level of performance and ease of use.

- Application deployers document lessons learned, best practices and script and package operation as part of the deliverables of the project.
Appendices
Cloud Community Platform Initiative
Introducing the Initiative

**Business Drivers**
- Growing pool of identified agency applications seeking cloud benefits
- 10 Geospatial and 4 general purpose applications
- Quantify savings, identify risks, ease migration

**Platforms**
- Construct a discrete subset of secure supporting platforms
- Develop processes for rapid application migration
- Leverage certifications and platform commonalities
- Develop chargeback mechanisms for self-sustaining funding

**Best Practices**
- Capture lessons learned
- Document techniques, issues and solutions
- Document cost models and benchmarks
- Build a cloud platform support community
- Maintain platforms, scripts and updates for client agencies

**Results**
Platform as a Service Architectural Framework
Distinguishing Application, Platform and Infrastructure

Applications
10 identified GeoSpatial Applications + more agency apps...

Platform

Application Servers
Platform Enablers (DB’s, etc)
App Frameworks / Libraries
Runtime Systems

Infrastructure
Operating System
Virtual Machines / CPU
Storage

Application Servers: Provide the deployment environment for actual business applications with access to enablers, frameworks and runtimes.

Platform Enablers: Platform enablers provide core supporting capabilities for developing, testing and deploying code, including DBMS, Directory, Queue, Service Bus, etc. A relational database is the most common enabler example but is not present in all platforms.

Frameworks provide bundled access to common behaviors and services, which applications can rely upon.

Libraries are reusable code modules that can be called directly from an application. Frameworks and Libraries save time and expense freeing developers from having to build common code and behaviors.

Runtimes: provide the execution support for developing and running the code. Examples include Java, Python, Microsoft Common Language Runtime, etc.