21st Century IT Architecture

OTN Architect Day San Francisco, July 9, 2013

Dr. James Baty, Vice President
Oracle Global Enterprise Architecture Program
21st Century IT Architecture

Abstract

• Applications are services running in the multi-tenant hybrid cloud. Companies beg their customers to tweet them, friend them, and publically rate their products. And constantly analyzing a deluge of Internet, social and sensor data is the key to creating the next super-successful product, or capturing an evil terrorist.

• The old IT architecture was planned, dedicated, stable, controlled, with separate and well defined roles. The new architecture is shared, dynamic, continuous, XaaS, DevOps.
The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle’s products remains at the sole discretion of Oracle.
Future IT
The end of the world as we know it

• The end of personal computers
  – Replaced by cellphones, cars, appliances

• The end of servers
  – Replaced by services

• The end of data centers
  – Replaced by the cloud

• The end of IT
  – Replaced by Infocommunibankertainment
Future Ops
– Hyper-integration, Radical Simplification
Clouds – nothing new, and yet…

Cloud Computing….
- It's about nothing new really (e.g., consolidation, automation).
- And yet it changes everything (10X+ scale & velocity).

- **Classic ‘data center’**
  - 100 servers
  - Install app + update once/yr
  = 200 events / yr

- **Cloud**
  - 10,000 ‘servers’
  - Deploy payload once/hr
  = 87,640,000 events/yr

*The Solution?*
*Use an ‘integration’ strategy to change the level of abstraction*
To Achieve Scale and Velocity – Transform Operations

– Factors leading to the Rise of Devops

- Increased usage of data center automation and configuration management tools
- Wide availability of virtualized and cloud infrastructure from internal and external providers
- Demand for an increased rate of production releases from application and business unit stakeholders

Integrate development & operations, Build self-service model, Move to continuous release
Move to Integrated Model / Service Management
– the ‘databases’ of devops

• **Model management** *(build-time)*
  – Exposes resources to developers
  – Stores developer’s models

• **Service management** *(run-time)*
  – Capacity management
  – Service management

• Developer’s main point of interaction with the cloud

• Defines services

• Includes solution catalog *(deployable entities)*

• Operator’s main point of contact for the cloud

• Provisions resources

• Includes configuration repository *(current state)*
Scale ops – raise abstraction / reduce complexity
Think ‘services’ and ‘PODs’

– Virtualization Centric
  • Potential of increasing complexity over time

VS.

– Integration Centric
  • Integration & architectural abstraction offers greater performance & ‘simplicity’
Mapping Models to Systems

Oracle Cloud – all key functions on Exadata / Exalogic
Future ‘enterprise computing’
– everyone becomes an SP
The SP shift – enterprise evolution

IT as a Support function

Enterprise Computing

Enterprise becoming Service Provider

Turning key apps into Public services

Internal Service Provider

Commercial Service Provider

IT as a business

All three need:
- multi-tenancy
- Federated identity
- Charge back billing
etc.

Where are you headed? How does this affect your technology choices?

Coordinated multi-division co. w/ many internal customers
Choices – Service Model
IaaS, PaaS or SaaS?

- You don’t have to build SaaS on PaaS (or PaaS on IaaS)
- Aspire for the highest level
Choices – Deployment Model
Private – Public – Hybrid – Managed - … ?

- Private Cloud
- Managed Cloud (Virtual Private Cloud)
- Public Cloud

Build
Managed
Subscribe

Build and manage your own cloud using cloud enabling products
Provider builds and manages a private cloud for you
Consume X as subscription-based services

• Most Enterprises – likely some mixture of all three
Beyond public vs private,  
– Use 4 dimensions to disambiguate offerings

**Who controls…**  
**Location, Architecture, Ownership, Operations …?**

<table>
<thead>
<tr>
<th>Who Controls?</th>
<th>Private Cloud</th>
<th>Colo VPC Cloud</th>
<th>Managed Cloud</th>
<th>Leased Appliances Cloud</th>
<th>Public Cloud</th>
</tr>
</thead>
<tbody>
<tr>
<td>Location</td>
<td>Enterprise ✔️</td>
<td>Enterprise ✔️</td>
<td>Enterprise ✔️</td>
<td>Enterprise ✔️</td>
<td>Enterprise ✔️</td>
</tr>
<tr>
<td>Architecture</td>
<td>Enterprise ✔️</td>
<td>Enterprise ✔️</td>
<td>Enterprise ✔️</td>
<td>Enterprise ✔️</td>
<td>Enterprise ✔️</td>
</tr>
<tr>
<td>Ownership</td>
<td>Enterprise ✔️</td>
<td>Enterprise ✔️</td>
<td>Enterprise ✔️</td>
<td>Enterprise ✔️</td>
<td>Enterprise ✔️</td>
</tr>
<tr>
<td>Operations</td>
<td>Enterprise ✔️</td>
<td>Enterprise ✔️</td>
<td>Enterprise ✔️</td>
<td>Enterprise ✔️</td>
<td>Enterprise ✔️</td>
</tr>
</tbody>
</table>

*They’re all ‘clouds’*
All future IT is about brokering Hybrid Clouds

Leverage multiple resource pools…. Over which to establish your value

Different Hybrids –

• **Functional Distribution** (designer)
  - Different components in separate clouds (e.g., CRM, HR)
  - Loosely coupled vs. tightly coupled

• **Lifecycle Distribution** (developer)
  - Separate development and test
  - Occasional vs Frequent release

• **Workload Distribution** (operator)
  - “Cloudbursting”
  - ACID vs (BASE)

Cloud Brokers –

manages use & delivery of clouds

• **Service Intermediation:**
  Provides value-added services
  e.g., unified access control

• **Service Aggregation:**
  Integrates multiple services
  e.g., data integration

• **Service Arbitrage:**
  Select from multiple Providers
  e.g., lowest cost

Leverage multiple resource pools…... Over which to establish your value
Future Apps
– everything is a service catalog
What *Forces* influence your Cloud Adoption?

<table>
<thead>
<tr>
<th>Business Drivers</th>
<th>Pure $ savings</th>
<th>vs.</th>
<th>Business Agility</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Technology Adoption</strong></td>
<td>Mainstream Adoption</td>
<td>vs.</td>
<td>Early Adoption</td>
</tr>
<tr>
<td><strong>Business Model</strong></td>
<td>IT as Support</td>
<td>vs.</td>
<td>IT as a Business</td>
</tr>
<tr>
<td><strong>Project Control</strong></td>
<td>IT Dept.</td>
<td>vs.</td>
<td>Business</td>
</tr>
</tbody>
</table>

*Is your project ‘cutting costs’ or ‘driving business’?*
Cloud Services & Service Models
A Business View on the Service Models

**Use Cases:** back office business processes that do not differentiate a business from its competitors, e.g. Email, Payroll, etc

**Benefits:** automation, agility

**Use Cases:** critical differentiated business processes deliberately custom built on top of selected technology platforms

**Benefits:** differentiation, automation, standardisation

**Use Cases:** re-hosting existing workloads (without change) or building highly differentiated platforms + applications

**Benefits:** consolidation, infrastructure automation

---

© 2013 Oracle

OTN Architect Day – 21st Century IT Architecture
PaaS Architecture

Which Multi-tenancy Implementation?

- Probably all three
The Service Catalog Drives Implementation

- Service Catalog has multiple dimensions
  - Availability SLAs (99.99)
  - Business Priority (24x7)
  - Regulatory issues (HIPAA)
  - Technical Versions
- This drives architecture / purchasing
- But make sure you know what dimensions are used
Map Service Catalog to

**PaaS / SaaS, Fully integrate Java apps and MW**

**Exalogic**  
*Best for Oracle & Java Apps*

**PaaS, High Performance Oracle DB, Only select options**

**Exadata**  
*Best for OLTP & DW*

**IaaS, any SW, any version**

**SPARC SuperCluster**  
*Best for DB & App Consolidation*
Big Data
– Social Analytics
the Data 'Tsunami'
– ingest, archive, distributed, fast, open.....

• 281 exabytes
• 45GB /person
• 10X growth in 5 years
• 50% thrown away
• excess 'halo' effect
• 1.1 MB email to 4 people
  - \( = 51.5 \text{ MB} \)
Big Data – the four Vs

• **Big data** – data sets so large and complex that they become difficult to process using on-hand database management tools or traditional data processing applications

• **Big data** – are high **volume**, high **velocity**, and/or high **variety** information assets that require new forms of processing to enable enhanced decision making, insight discovery and process optimization.

• Plus ‘veracity’…
Do you have an Analytics Pyramid?
The Value from Big Data

Industry movement towards

- strategic,
- integrative,
- competitive

analysis

Competitive analytics
Strategic, Integrating external data

Business focused
Cross LOB or company wide

Focused projects
Localized analytics

Basic visibility
Internal reports

Mostly missing
Consider Customer Experience
LOTS or Data, Transactional, Social, Contextual

<table>
<thead>
<tr>
<th>NEED / RESEARCH</th>
<th>SELECT</th>
<th>PURCHASE</th>
<th>RECEIVE / USE</th>
<th>MAINTAIN / RECOMMEND</th>
</tr>
</thead>
<tbody>
<tr>
<td>WEB</td>
<td>Comparison Site</td>
<td>Order Online</td>
<td>Pickup Local Store</td>
<td>Order Online</td>
</tr>
<tr>
<td>CONTACT CENTER</td>
<td>Visit Retail Store</td>
<td>Chat</td>
<td>Call for Info about Add-on Accessories</td>
<td></td>
</tr>
<tr>
<td>IN-STORE</td>
<td>Kiosk</td>
<td>Change Order</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CATALOG</td>
<td>Product Info</td>
<td>Select Product</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MOBILE</td>
<td>Web Search</td>
<td>Email Order Confirm w/Rec</td>
<td></td>
<td></td>
</tr>
<tr>
<td>EMAIL</td>
<td>Ask Facebook Friends For Recommendations</td>
<td>Read Reviews</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SOCIAL</td>
<td></td>
<td></td>
<td>Tweet About Purchase Experience</td>
<td>Ask for Help on Community Chat Room</td>
</tr>
</tbody>
</table>
Social Services – for Analytics
A top level Service Category in Oracle Cloud
Future Architecture
– Use cases, patterns, ecosystem
Common Early Cloud Use Cases
– architectural encapsulation and re-use

- **Test and Development**
  - (Batch Processes – TimesMachine)

- **Functional Offload**
  - (Storage – SmugMug)

- **Augmentation**
  - (Temporary Load – Animoto)

- **Web Service**
Design patterns – e.g., load balancing – from scripts to appliances and back

- Round Robin DNS (& lbnamed)
  - simple scripting
  - 'virtual appliance”
  - 1995

- Cisco Local Director (& Resonate)
  - Basic appliance
  - 'pc' + SW
  - 1997

- F5 Big-IP (& Arrowpoint)
  - Built into 'network'
  - Wire speed + SSL
  - 2000+

- mod_backhand wackamole
  - Basic appliance
  - Back to scripts
  - Cheap super scale
  - 2008

Same design pattern... distribution, load balancing, failover, scalability... – but several different implementations of the pattern – some 'appliances', 'some virtual appliances'
• We used to think in analogies of individual things – 'blueprints'

• Start thinking in analogies for ecosystems – 'urban planning'

• Skills - refactoring, use cases, design patterns, ...