On Demand Anwendungen
und ihre Anforderungen an Datenbanksysteme

Rudolf Munz
SAP AG
Mai 2010
Agenda

1. Characteristics of On Demand Applications
2. On Premise vs. On Demand Applications
3. Provider’s Perspective on On Demand Applications
4. Multi-Tenancy
5. Extensibility
6. Encryption
7. Business Continuity
8. Conclusions
1. Characteristics of On Demand Applications
2. On Premise vs. On Demand Applications
3. Provider’s Perspective on On Demand Applications
4. Multi-Tenancy
5. Extensibility
6. Encryption
7. Business Continuity
8. Conclusions
The NIST Definition of Cloud Computing

Five essential characteristics:

- **On-demand self-service** provisioning of computing capabilities to end consumers
- **Broad network access** through standard mechanisms from any device
- Dynamic and location-independent **resource pooling**
- **Rapid elasticity** of resource provisioning for (virtually unlimited) computing capabilities
- **Measured services** by a metering capability to control and optimize resources

Three service models:

- **Infrastructure as a Service (IaaS)**
- **Platform as a Service (PaaS)**
- **Software as a Service (SaaS)**

Four deployment models:

- **Public cloud** is made available to the general public
- **Community cloud** is shared by several organizations
- **Private cloud** is operated solely for an organization
- **Hybrid cloud** is a combination and cooperation of private, community, or public cloud entities
Infrastructure as a Service (IaaS)

Example:
http://aws.amazon.com
Platform as a Service (PaaS)

Example:
http://code.google.com/intl/de/appengine/
Software as a Service (SaaS)  
On Demand Applications

Example:  
http://www.salesforce.com/de/
Impact of Cloud Computing

Expectations:

- The impact of cloud computing is reminiscent of the shift from big mainframe systems to client server computing in the '90s

- A steep change in the economics of computing is creating opportunities for new providers, while threatening the established business models of traditional application software and technology vendors
Deployment Model and Business Model
On Premise vs. On Demand

<table>
<thead>
<tr>
<th>Deployment Model</th>
<th>Customer Site</th>
<th>Provider Site</th>
</tr>
</thead>
<tbody>
<tr>
<td>On Premise</td>
<td>On Premise</td>
<td>Application Hosting</td>
</tr>
<tr>
<td>Managed Appliance</td>
<td>SaaS On Demand</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Business Model</th>
<th>Upfront Licenses</th>
<th>Subscription</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Characteristics of On Demand Applications (1) (Customer’s Perspective)

**Ease of consumption**
- On Demand trials and sales (discover, try, buy)
- Short learning curve to get started (i.e. minimal training required)
- Simple and intuitive user interface
  - Allows discovery of the functionality
  - Enables productive use by experts
- Non-disruptive delivery of new functionality (in many small increments)

**Ease of deployment**
- Requires only an Internet browser and a user account
- No configuration (runs out-of-the box)
- Configuration capabilities are optional

**Outsourcing**
- The provider takes care of installation, patches, upgrades, performance, availability, software and hardware platforms, backup / recovery
- Favors buying decision by the line of business (often bypassing central IT)

**Subscription**
- Monthly subscription fees per user account (pay as you go)
- Predictable costs
- Minimal customer commitment and risk
### Characteristics of On Demand Applications (2) (Customer’s Perspective)

| Simple applications | Simple and standard applications with an identical code base |
| Simple setup         | No need for massive data migrations |
| Weak integration     | Mashups with other On Demand applications are simple |
| Good extensibility   | Personalization |

- Use the product as is (i.e. with the present feature set)
- No modifications allowed (only extensions)
- Good extensibility by customers and partners
- Data are typically entered via the On Demand application
- Existing data can also be entered by uploading Excel and CSV files
- Data can be checked out (e.g. at subscription end)
- Integration into customer’s On Premise IT infrastructure is a challenge
- Integration should not be a prerequisite for On Demand applications
- Configurable user interface (e.g. simplification)
- Options to switch on additional functionality
- Structural extensibility (e.g. add a new field from UI to persistency)
- Behavioral extensibility (e.g. add new logic)
- Extensibility tools (power user) and extension coding (developer, partners)
Agenda

1. Characteristics of On Demand Applications
2. **On Premise vs. On Demand Applications**
3. Provider’s Perspective on On Demand Applications
4. Multi-Tenancy
5. Extensibility
6. Encryption
7. Business Continuity
8. Conclusions
On Demand vs. On Premise Applications

- Integration
- Extensibility
- Criticality
- Operation costs
- Security
- Standardization

On Premise vs. On Demand Applications
## On Demand vs. On Premise (1)

| Standardization | Differentiation versus standardization of a particular application  
|                 | Standardization favors On Demand applications |
| Extensibility   | Are extensions or modifications needed?  
|                 | Extensibility requirements favor On Demand applications  
|                 | The need for modifications favors On Premise applications |
| Operation costs | Is the total cost of ownership (TCO) dominated by operation costs or by extensions or modifications of the application?  
|                 | Updates / upgrades of On Demand software are non-disruptive  
|                 | Disruptive upgrades of On Premise software add to operation costs  
|                 | High operation costs favor On Demand applications |
On Demand vs. On Premise (2)

**Criticality**
- Is the application mission-critical?
- What costs are associated with its downtime and outages?
- What service level agreements (SLA) are offered by the On Demand provider?
- High degree of criticality (presently) favors On Premise applications

**Security**
- Confidentiality and privacy of data, legal requirements
- Isolation of tenant data (in a multi-tenant environment)
- High security requirements favor On Premise applications

**Integration**
- Is read or write integration into the On Premise IT systems required?
- Efforts for integration project vs. operation cost savings
- Integration middleware either at the customer’s or at the provider’s site
- Deep integration requirements favor On Premise deployment
Agenda

1. Characteristics of On Demand Applications
2. On Premise vs. On Demand Applications
3. Provider’s Perspective on On Demand Applications
4. Multi-Tenancy
5. Extensibility
6. Encryption
7. Business Continuity
8. Conclusions
Different Value System for On Premise and On Demand Applications

On Premise Applications

- New Features
- Reduce TCO
- Sales, Support, Development Effectiveness

On Demand Applications

- Reduce Total Cost of Operations
- New Features
- Sales, Support, Development Effectiveness
## On Demand Application Characteristics (Provider’s Perspective)

<table>
<thead>
<tr>
<th>Online sales</th>
<th>Minimal operation costs per user</th>
<th>Immediate feedback</th>
</tr>
</thead>
</table>
| - Online sales process (discover, try, buy)  
- Quick provisioning of trial and productive accounts required  
- Sales offering is continuously improved by trial feedback | - High initial software quality  
- Identical application for all customers (tenants)  
- Single codeline, i.e. all customers use the same version (same executable)  
- Common metadata, e.g. common SQL data model  
- Tenant-specific data (data isolation)  
- Tenant-specific extensions (based on tenant-specific metadata and data)  
- Economy of scale for a large number of tenants using the same infrastructure  
- Large tenants on dedicated systems (single tenant systems)  
- Small tenants require multi-tenant systems | - Providers can easily collect and analyze data about the application usage  
- Usability improvements by clickstream analysis  
- Trial of new features (e.g. in 5% of the user population)  
- Instant feedback for development and support  
- Customer voting on development priorities |
The On Demand Business Model
(Provider’s Perspective)

Subscription Revenue

Cost of Development

Total Cost of Operations

Cost of Sales

Profit

COST COMPONENTS

Support

Application Management

Infrastructure Management

Number of customers

Number of systems

- X

- X

X

- X
Sharing Concepts for On Demand Applications

Single tenant system
- Each customer has an own system running on dedicated hardware
- Maximum flexibility by full individualization of the system
- Individual service level (SLA) including maintenance windows

Multiple systems on one hardware
- Multiple systems share one hardware by virtualization techniques
- Better hardware utilization without impact on application systems
- Full tenant isolation down to the operating system
- Dedicated application system per tenant

Multi-tenant application systems
- Multiple tenants share one application system
- System related costs are reduced according to the number of tenants
- Fixed assignment of tenants to a particular application system
- Example: 100 tenants with about 25 users in one application system
Cloud Concept for On Demand Applications

Flexible Assignment of Tenant Content to Tenant Containers

Storage Pool with Tenant Content

© SAP 2010 / Page 21
Provider Operations at the Tenant Level

Tenant creation / deletion

Tenant startup / shutdown

Tenant move (for load balancing)
- The speed of a tenant move matters, slow tenant move leads to overprovisioned systems (expectation: hours → minutes → online)

Tenant copy (for test tenants)
- The same customer can use several tenants (e.g. for different legal entities, productive tenant and test tenants, software logistics between tenants required)

Tenant export (for data migration)

Flexible assignment of tenants to application systems (at tenant startup time)

No tenant-specific backup. Only backups at the system level.

No point-in-time recovery at the tenant level, but fallbacks to previous tenant copies
Provider Operations at the System Level

System installation / de-installation

System startup / shutdown

System administration
- Patches, updates (OS, DBMS, application)
- Service packs (OS, DBMS, application)
- Release upgrades (OS, DBMS, application)
- Database backup / recovery
- No point-in-time recovery at the (multi-tenant) system level
- Load monitoring and leveling
- Tuning

Landscape administration
- Patches, updates (OS, DBMS, application)
- Service packs (OS, DBMS, application)
- Release upgrades (OS, DBMS, application)
- Database backup
- Load monitoring and leveling
Agenda

1. Characteristics of On Demand Applications
2. On Premise vs. On Demand Applications
3. Provider’s Perspective on On Demand Applications
4. **Multi-Tenancy**
5. Extensibility
6. Encryption
7. Business Continuity
8. Conclusions
Content Types

Provider Content

- All data which are under the control of the application provider
  (source code, byte code, executables, configurations, templates, ...)

System Content

- All runtime data of a particular application system
  (byte code, machine code, trace, log, monitoring data, ...)

Tenant Content

- Tenant-specific application data
- Tenant configuration and customizing data
- Tenant-specific runtime data (e. g. messages in queues)
- Tenant extensions

Axioms

- Tenants do not change Provider Content
- Provider does not change Tenant Content
Content Ownership and Lifecycle Management

Provider Delivery
(with Provider Lifecycle Management)

Initially Empty
(without Provider Lifecycle Management)

Tenant-independent

Tenant-dependent

Provider Content

System Content

Tenant Content
Multi-Tenancy Support at the Database Layer (Alternatives)

Dedicated database per tenant
- Tenant isolation becomes simple
- Tenant extensibility becomes simple
- High cost of operations (many systems with backup / recovery, monitoring, tuning, patches, …)
- Only suited for a population of tenants with a large number of users

Dedicated schema per tenant
- Separate schema (set of tables) per tenant
- Tenant isolation by tenant-specific schema visibility (default schema per database user)
- Allows physical separation of tenant content
- Additional tenant-independent schema is required for provider and system content
- Multiple schemas enable tenant-specific schema extensions

Dedicated rows per tenant
- Shared tables contain rows of different tenants (primary key is pre-fixed by Tenant-ID)
- Tenant isolation by tenant-specific views
- Allows physical separation by table partitioning according to Tenant-ID
- Additional tenant-independent tables are required for provider and system content
- Single schema requires tenant-specific metadata separation (e.g. by extension tables)
### Dedicated Rows per Tenant

#### Logical and Physical Data Separation

#### Tenant Partitions

<table>
<thead>
<tr>
<th>Tenant-ID</th>
<th>Column1</th>
<th>Column2</th>
<th>Column3</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>11</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>11</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>12</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>12</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>12</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>13</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>13</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>13</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>13</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

#### Tenant Volumes

- Tenant 11
- Tenant 12
- Tenant 13
Multi-Tenant Infrastructure

- Application Server
  - Application Server Sessions
  - Application Server Caches

- Database Server
  - Common SQL Catalog
  - Database Sessions
  - Database Cache

- Storage System
  - Application Content
  - System Content
  - Tenant X Content
    - Tenant X Extensions
  - Tenant Y Content
    - Tenant Y Extensions
  - Tenant Z Content
    - Tenant Z Extensions

- Tenant-independent

- Tenants: Tenant X, Tenant Y, Tenant Z
1. Characteristics of On Demand Applications
2. On Premise vs. On Demand Applications
3. Provider’s Perspective on On Demand Applications
4. Multi-Tenancy
5. **Extensibility**
6. Encryption
7. Business Continuity
8. Conclusions
Field Extensibility

Tenant uniformity by
- Identical application coding for all tenants (common executable / bytecode)
- Identical table structures for all tenants (common SQL catalog)

Tenant uniformity collides with tenant extensibility

Extensions (e.g. UI screens, code, structures) must become tenant-specific metadata

Option A - generic implementations:
- Pre-allocated spare fields (e.g. 50 numeric and 50 string fields) in each table. Tenant-specific usage is described by metadata table.
- Name-value pairs in generic extension table (Tenant-ID, table, key, attribute, type, value)
- Assessment: Both approaches lack scalability with respect to the number of users and table sizes

Option B - specific implementations:
- Tenant-specific extension columns (only possible for dedicated database or schema per tenant)
- Tenant-specific extension columns in tenant-specific extension table
- Assessment: Good exploitation of database functionality
Extensibility at the Persistency Layer (Instance Data)

Separate extension tables for tenant-specific field extensions

<table>
<thead>
<tr>
<th>Tenant 1</th>
<th>Tenant 2</th>
<th>Tenant N</th>
</tr>
</thead>
<tbody>
<tr>
<td>![Table T](A B C) ![Extension Table T1](D E)</td>
<td>![Table T](A B C) <img src="F" alt="Extension Table T2" /></td>
<td>![Table T](A B C) ![Extension Table TN](G H)</td>
</tr>
</tbody>
</table>

...
Tenant and Namespace Concept

Customer Namespaces
(including non-invited development partners)

Application Provider Namespaces
(including invited development partners)

Namespace concept is needed to express ownership

Multi-Tenant concept is needed for tenant isolation
Extensibility at the Database Layer (Metadata)

- Customer namespaces
- Non-invited partner namespaces
- Invited partner namespaces
- Application Provider namespaces

Tenant-independent

Tenant 1
- Extension Catalog (new tables)

Tenant 2
- Extension Catalog (new tables)

Tenant N
- Extension Catalog (new tables)

Common SQL Catalog

Namespaces for table names are required to express ownership.
Tenant Isolation and Namespace Ownership

- Customer namespaces
- Non-invited partner namespaces
- Invited partner namespaces
- Application Provider namespaces

Tenant-specific visibility of non-invited partner extensions

<table>
<thead>
<tr>
<th>Tenant</th>
<th>Tenant 1 (Customer namespace)</th>
<th>Tenant 2 (Customer namespace)</th>
<th>Tenant N (Customer namespace)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Many non-inv. partner namespaces</td>
<td>Many non-inv. partner namespaces</td>
<td>Many non-inv. partner namespaces</td>
</tr>
<tr>
<td></td>
<td>Many inv. partner namespaces</td>
<td>Many Provider namespaces</td>
<td></td>
</tr>
</tbody>
</table>
Agenda

1. Characteristics of On Demand Applications
2. On Premise vs. On Demand Applications
3. Provider’s Perspective on On Demand Applications
4. Multi-Tenancy
5. Extensibility
6. Encryption
7. Business Continuity
8. Conclusions
## Encryption Infrastructure

<table>
<thead>
<tr>
<th>Vulnerability</th>
<th>Mitigation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Communication via Internet</td>
<td>Encryption via https</td>
</tr>
<tr>
<td>Illegal copies of tenant data by provider’s operation team</td>
<td>Encryption of tenant volumes, encryption of backups</td>
</tr>
<tr>
<td>Attacks to tenant data by other tenants</td>
<td>Encryption of tenant volumes</td>
</tr>
<tr>
<td>Illegal access to tenant data by provider’s support team</td>
<td>Field-level encryption of sensitive data (e.g. credit card numbers, social security numbers) at the UI</td>
</tr>
</tbody>
</table>

- System-level encryption of backups (data and log)
- Tenant-level encryption of tenant data
- User-level encryption of communications
- Field-level encryption (within the browser) of sensitive data
1. Characteristics of On Demand Applications
2. On Premise vs. On Demand Applications
3. Provider’s Perspective on On Demand Applications
4. Multi-Tenancy
5. Extensibility
6. Encryption
7. **Business Continuity**
8. Conclusions
Business Continuity must be supported by all On Demand infrastructure layers (application, application server, database server, operating system, hardware)

This means to eliminate planned downtime for

- Configuration changes (online or rolling)
- Patches of executables (rolling)
- Patches of byte codes (online or rolling)
- Update / upgrade to new functionality (make new functionality switchable)
- Update / upgrade to extended persistency layer (with new tables and new fields)
  - Mirror system with capture and replay of changes (initial plus delta)
  - Support of application tests on upgraded mirror system (with fallbacks)
1. Characteristics of On Demand Applications
2. On Premise vs. On Demand Applications
3. Provider’s Perspective on On Demand Applications
4. Multi-Tenancy
5. Extensibility
6. Encryption
7. Business Continuity
8. Conclusions
DBMS Requirements

Logical tenant data separation (dedicated schema or dedicated rows)

Identical application and SQL coding for all tenants

Physical tenant data separation (tenant volumes with automatic space management)

Online move of tenant volumes between database instances

Encryption of tenant volumes and backups

Online “everything“ (i.e. configuration changes, patches, upgrades, …)

Flexible assignment of tenant volumes to database instances at tenant startup time

Extensibility of “dedicated rows per tenant“ approach benefits from:

- Extension tables with implicit join behavior
- Namespace support in table names
Conclusions

On Demand apps are a new type of applications – not just a new deployment model.
They deliver simple and standardized applications for customers with little IT skills.
Buying decisions move from the central IT department to the lines of business.
On Demand apps will not replace On Premise apps.
Integration of On Demand apps with On Premise apps remains a challenge.
Integration of On Demand-only or On Premise-only apps is simpler.
Attractive features of On Demand apps will raise the bar for On Premise apps.
On Demand apps require a multi-tenant infrastructure with business continuity.
Tenant uniformity and tenant-specific extensibility are conflicting goals.
Thank you!