



# DataFlex Entwickler Tag

## 20 November 2014

Frankfurt, Germany

# MultiTenancy Applications

Presentation by **Eddy Kleinjan**

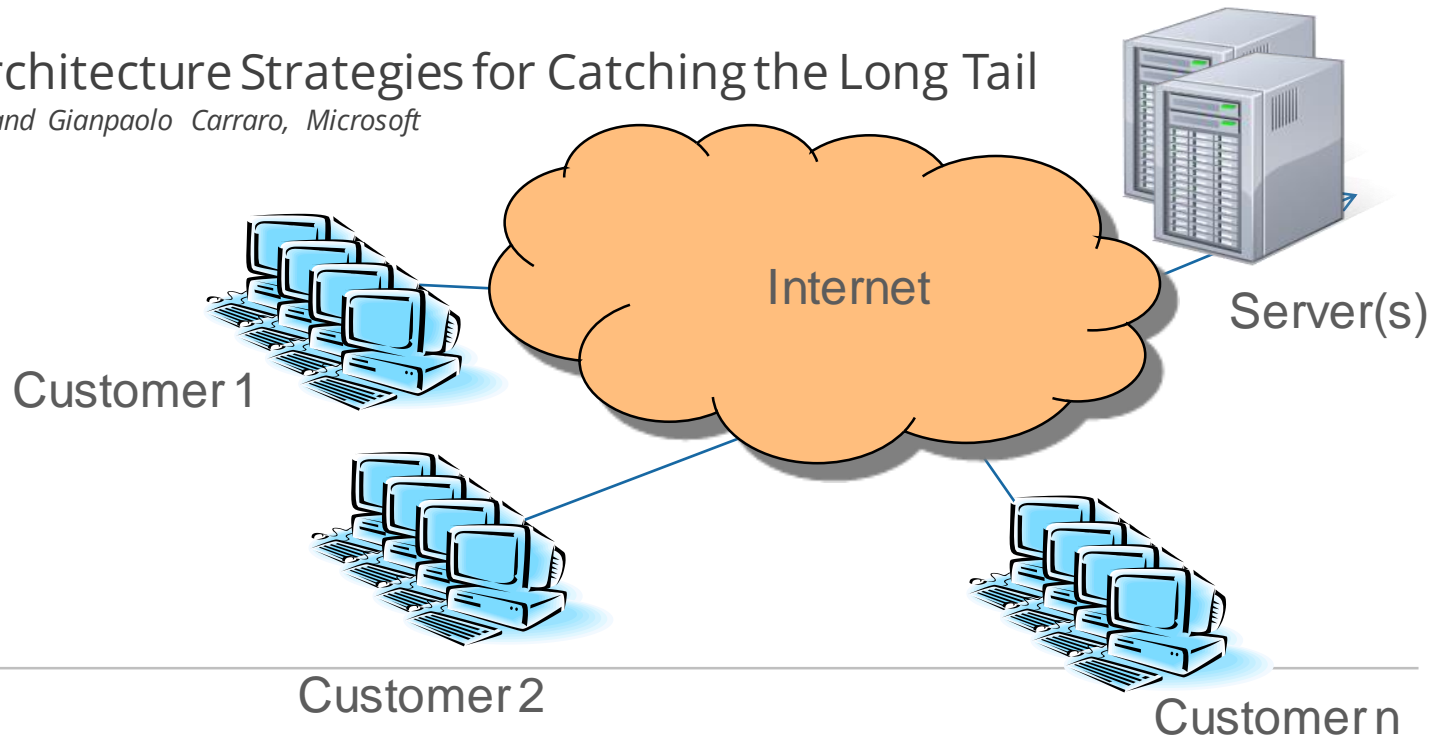
Going into the cloud sooner or later brings up how to handle multi-tenancy in applications.

*Are you ready to take your application to the next level?*

# SaaS

- Software as a Service
- **"Software deployed as a hosted service and accessed over the Internet."**
- Source: Architecture Strategies for Catching the Long Tail

*Frederick Chong and Gianpaolo Carraro, Microsoft*



# Reduce Costs

- Changing the Business Model
- Shifting "ownership" of the software from the customer to an external provider
- Reallocating responsibility from customer to provider
  - Software
  - Hardware
  - Professional Services
- Reducing cost

# Business Models

- Outsource IT Hardware and Software to Service Providers reducing cost
- Reduce up-front costs
- Single Version of the Software
- Applies to both WebApps and WinApps
- Charging metrics:
  - Users
  - Tenant
  - Item (customer, invoice, etc.)
  - Time (Day, Month, etc.)

# Trust

- Who is the service provider?
- Where is the data stored?
- What happens when one of the goes broke?
- Who owns the data
- Who takes care of back-up?
- Can I move my data out when leaving?
- Etc...

# Attributes of a Multi-Tenant Architecture

Configurability

Multi-Tenancy

Scalability

# Scalability

- Process Pooling (Scale Up)
- Load Balancing (Scale Out)
- SPLF:  
Security, Performance, Load Balancing and Failover
- Use Client / Server Database Model
- IP Address Sharing
- Multiple Servers
- Database Partitioning
- Bandwidth

# Multi-Tenancy

Ad Hoc / Custom

Configurable

Configurable,  
Multi-Tenant-Efficient

Scalable,  
Configurable,  
Multi-Tenant-Efficient



# Database Considerations

# How to design your database

# All data in one database

## Pros

- One database structure; easily maintained
- Create Cross-Tenant Queries Easily
- One backup procedure

## Cons

- Size of Database can grow
- Harder to Partition Data
- Careful To Separate Data At All Times
- Harder to restore a database of a Tenant

# Each Tenant in Separate Database

## Pros

- Data Can Easily Be Taken Out
- Databases can be partitioned over multiple database servers
- Physical Separation of data may be a Requirement
- Database Accounts to Control Access
- Backup and Restore databases individually

## Cons

- Structure Changes Need To Be Rolled Out Over Each Database
- Backup and Restore databases individually

# Process / Application Considerations

- One version of the software
- One WebApp Application Process or Separate process for each Tenant
  - Use WebApp Administrator to setup separate Application per Tenant
- Same URL for each Tenant or Separate

# Separate the data in WebApp

- Use Procedure OnAttachProcess as a hook to set Constraints

```
{ MethodType=Event }  
Procedure OnAttachProcess  
End_procedure
```

## Description

When process pooling is used, this event is sent to the WebApp object. By default, it does nothing, but it can be used to perform any initialization for the newly attached process.

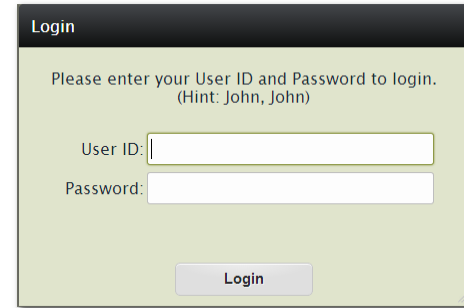
- Use DF\_FILE\_SQL\_FILTER to set a hard constraint
- When using psSQLFilter, mind pbApplyGlobalSQLFilters!!!
  - It's false by default, overruling DF\_FILE\_SQL\_FILTER

# Encryption

- Use HTTPS / SLL for traffic encryption
- Enforce it!

# Authentication

- User Id / Password
- Client Certificates
- Client Certificates for Smartcard
- IP Address
- Multi-Pass Authentication
  - Something you know
  - Something you have



The screenshot shows a web-based login interface. At the top, there is a dark header bar with the word "Login" in white. Below this, the main area has a light green background. It contains the instruction "Please enter your User ID and Password to login." followed by a hint "(Hint: John, John)". There are two input fields: "User ID:" and "Password:". Below these fields is a grey button labeled "Login".



# Authorization

- Manage Permissions
- Logging of access using Sessions
- Menu System
- View Authorization

# Consider

- Legal Aspects
  - Where is the data located?
  - Who is responsible for data loss?
- SLA (Service Level Agreement)
  - Guaranteed up-time
  - Monitor Server Availability
  - Monitor Server Performance

# Where to go from here?

- [Architecture Strategies for Catching the Long Tail](http://msdn.microsoft.com/en-us/library/aa479069.aspx)  
<http://msdn.microsoft.com/en-us/library/aa479069.aspx>
- [Multi-Tenant Data Architecture](http://msdn.microsoft.com/en-us/library/aa479086.aspx#mlttntda_tde)  
[http://msdn.microsoft.com/en-us/library/aa479086.aspx#mlttntda\\_tde](http://msdn.microsoft.com/en-us/library/aa479086.aspx#mlttntda_tde)
- Amazon RDS (Relational Data Services)
  - MySQL, Oracle, PostgreSQL
- Microsoft Azure SQL Database
  - Database as a Service

# Thank you.

Have a nice day!