LEARNING

LIVES HERE

Microsoft®

tech·ed

North America | 2009

LEARNING LIVES HERE

Software as a Service in the Cloud

Michael Stiefel
President
Reliable Software, Inc.

ARC 311

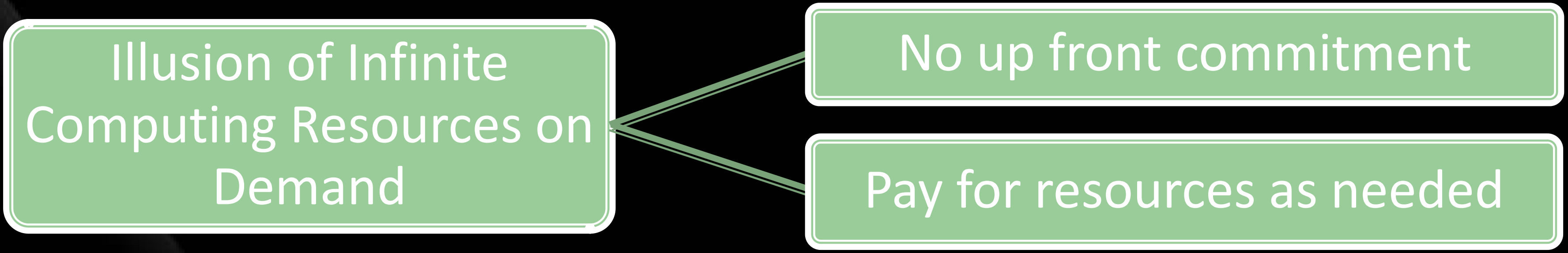
Cloud Computing is yet another technology revolution.

A case study will illustrate:

Key Issues and Opportunities for Cloud Computing

Cloud Computing can make the world a safer place

Cloud Computing is Utility Computing



Illusion of Infinite Computing Resources on Demand

No up front commitment

Pay for resources as needed

UC Berkeley Reliable Adaptive Distributed Systems Laboratory

Session Focal Points

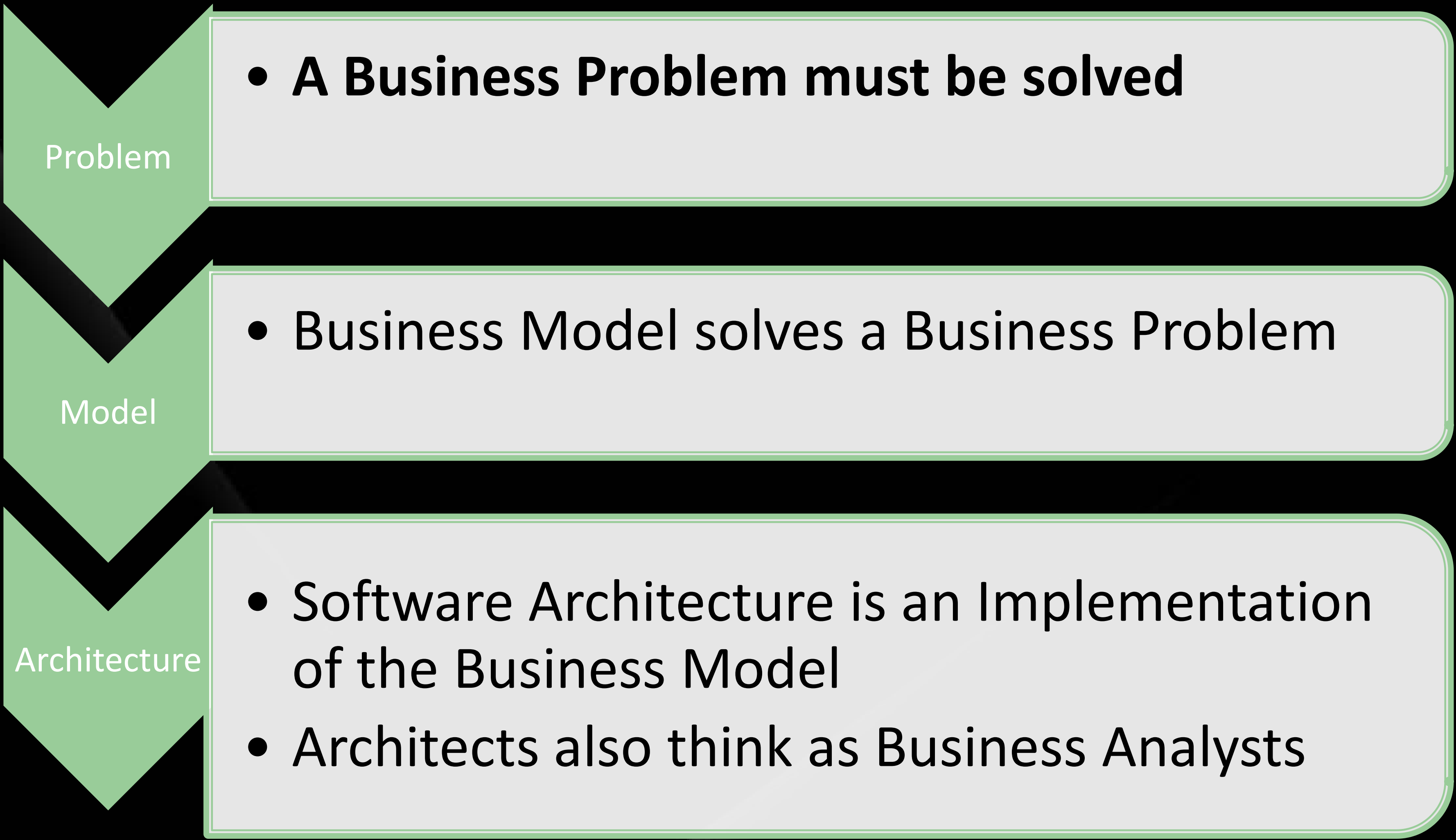
- Business Model drives Software Architecture
- Currently more relevant to SMB than Enterprise
- Magnifies Classic architectural and design issues
- Move to Windows Azure

Case Study

iTNAmerica®

Dignified transportation for seniors

Business Model Drives Architecture



Problem

- **A Business Problem must be solved**

Model

- Business Model solves a Business Problem

Architecture

- Software Architecture is an Implementation of the Business Model
- Architects also think as Business Analysts

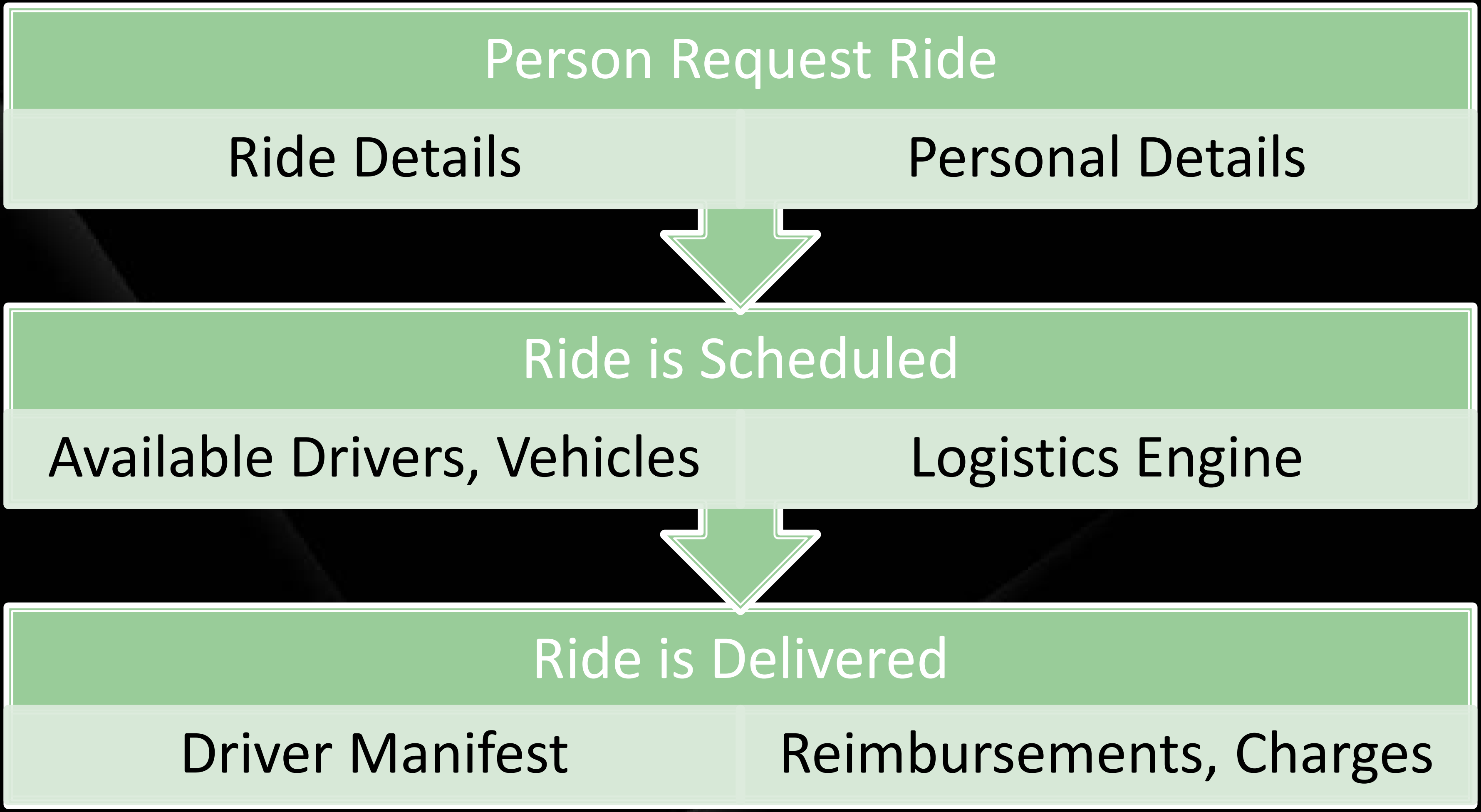
Elder Transport is a Major Social Issue

- People are outliving their ability to drive a car
- Senior population is growing
- Rides when they want and where they want.
- Seniors drive, endanger themselves and others
- Doctors would tell patients to stop driving if there was an alternative

Business Model Solves a Business Problem

- Riders pay for service
- Local affiliates
  - Local Drivers
  - Local Fund Raising
- Sufficient population density
- ITNAmerica provides technology and support

Ride Request Use Case



Person Request Ride

Ride Details

Personal Details

Ride is Scheduled

Available Drivers, Vehicles

Logistics Engine

Ride is Delivered

Driver Manifest

Reimbursements, Charges

First Solution

- Monolithic VB6 Application
  - SQL Server per affiliate
  - Accessed through Terminal Server
- Validated Business Model
- Problems
  - Does not scale for national and international rollout.
  - Cannot integrate with third parties.
  - Not a platform for other solutions such as rural transportation.

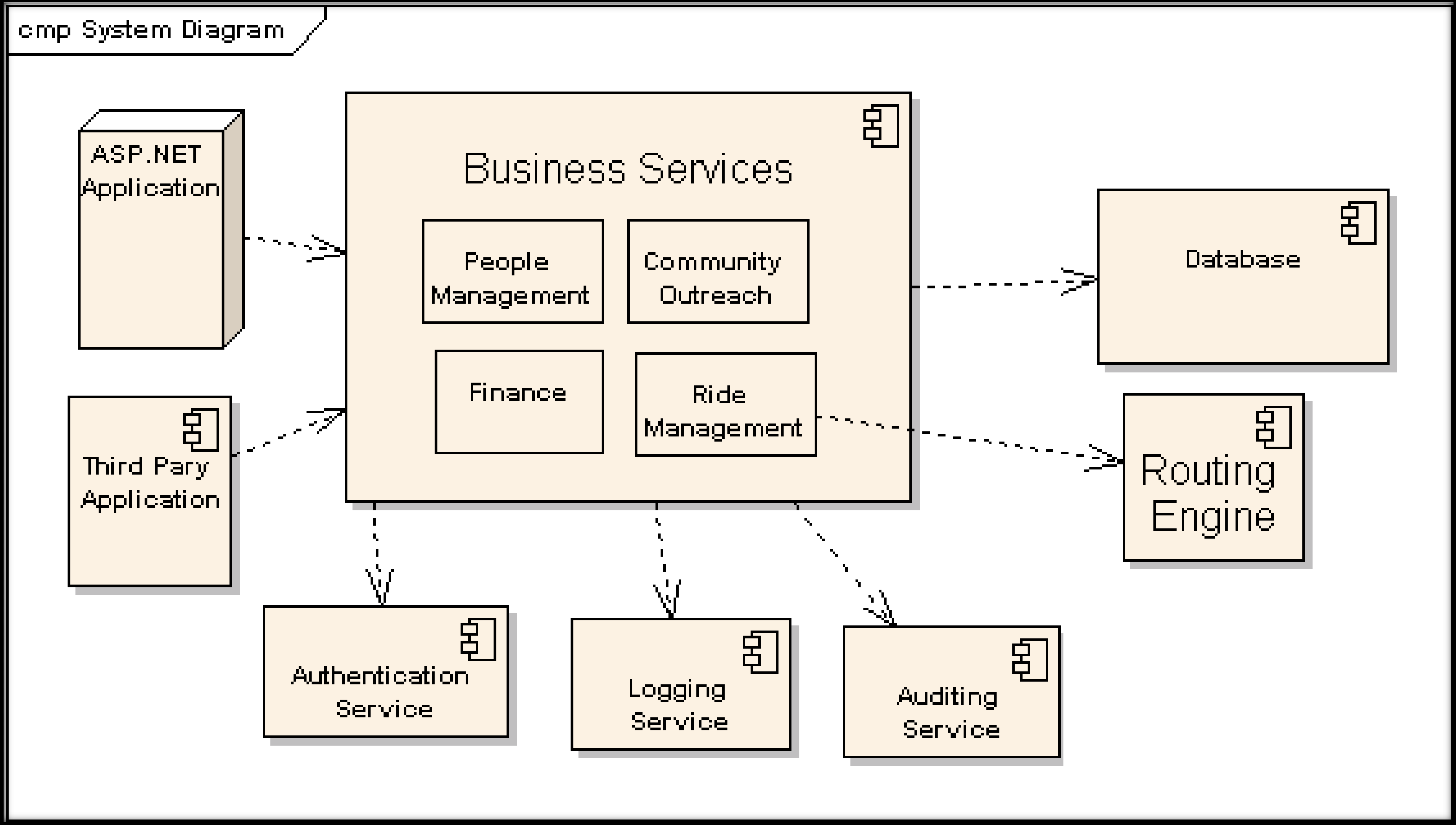
Application Options

- Traditional On-Premises Application
  - Desktop
  - Client / Server
- Off-Premises Application
  - Private Cloud / Self Hosted
  - Public Cloud

Architecture Implements Business Model

	Desktop	Web App	Web Service
Affiliates cannot maintain Infrastructure		x	x
Avoid installation, upgrade issues		x	x
Customers, Drivers use system anywhere		x	x
Portal Interop through Domain Layer			x
Third Party Interop through Domain Layer			x
Affiliate build, enhance offering			x
Platform for future offerings			x
Continual model validation		x	x
Protect Logistics and other IP		x	x

New Solution = Web App + Services



Mission Critical Application

- Ride delivery failure can mean death or disability
- Continually Validate Business Model
- Measure Rides, Not packets or updates
- Building a virtualized, private cloud

SMB Can Make the Impossible Possible

- Cheaper to deliver solution to customers
  - Scale to large number of users without complications of desktop support
  - Easier to upgrade clients to latest version
- Integrate with third parties to enhance solution
- Extend reach internationally
- Protect intellectual property

Architecture and Design Practices

Familiar Design Principles, but…

- Cloud is Different from on-premises application
- You do not control the network
  - Network Latency / Network responsiveness
  - Connectivity Loss is a Problem

Messages Across the Internet

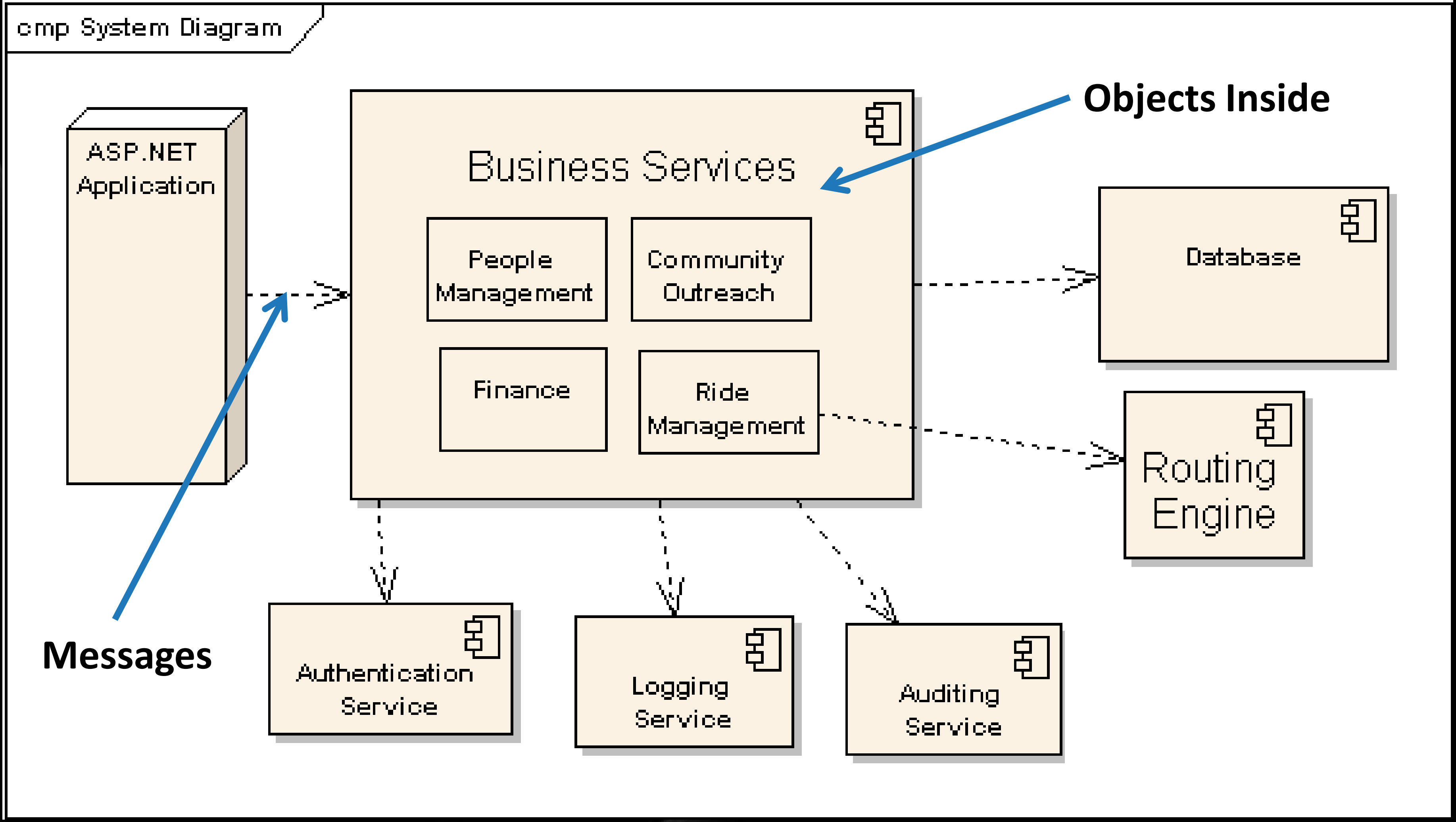
- Distributed objects across Internet will not scale
- Message is a discrete unit of business
    - Ride Request, Payment, Membership Application
    - New Membership = Membership Application + Payment

Object Orientation vs Messaging

```
Class Person
{
  public string Name {get; set;}
  public string Address {get; set;}
  public bool ValidateName();
  public bool ValidateAddress();
}
```

```
<MembershipApplication>
  <Name>Peter Jones</Name>
  <Address> 8500 Sunset Blvd. West Hollywood, CA </Address>
  <MembershipType>Donor</MembershipType>
  <WhoRecommended>Medical Office Brochure</WhoRecommended>
</MembershipApplication>
```

Where do objects belong?



Objects Inside

Messages

Service Tier

- Building stateless services, allow partial failure
- Domain objects do not last beyond message call (Unit of Work pattern)
- ACID transactions within service call, compensation across several messages

Service Definition

```
[ServiceContract(Namespace = "http://test.org/test/v1")]
public interface IService
{
  [OperationContract]
  [FaultContract(typeof(ServiceFault))]
  Response SaveMembership(Request request);
  …
}
```

Objects in the Implementation

```
public Membership SaveMembership(Request request)
{
  Response = new Response();
  try
  {
     Membership m = new Membership();
     MapToDomain(request.Membership, m);
     Facade facade = new Facade();
     facade.Save(m);
     facade.Flush();
     Update(response);
   }
   catch (...)
   {
      response.error = …
   }
   return response;
}
```

Web Application Tier

- Separate widgets from the application.
- Access business services via messages, through a façade layer.
- Compose business scenarios with multiple service calls.

Membership UserInterface

```
 public interface IMembership
 {
   string Name { get; set; }
   string Salutation { get; set; }
  …
 }

public class EditMembership : BaseControl
{
  …
  public void OnSave() // called from UI widget
  {
     UIFacade façade = new UIFacade();
     IMembership im = GetMembershipInfo();
     façade.SaveMembership(im);
  }
   …
}
```

Service Façade Pattern

```
 public class UIFacade : IFacade
 {
…
  public public bool SaveMembership(IMembership im)
  {
    ServiceClient client = new ServiceClient();
    IWebSecurity ws = WSecurity.Get();
    ws.AddCredentials(client);
    Request request = new Request();
    PopulateRequest(request, im);
    Response response = client.SaveMembership(request);
    PopulateUI(im, response);
    …
  }
```

Database Tier Choices

- Tenancy
  - Multiple tenants in one database, tenant id column
  - One tenant per database
- Schema and Customization
  - Schema per tenant, customize schema
  - Single schema
    - Uniform data model across tenants, data driven
    - Metadata or XML driven customization
    - Reserved Columns

Problems of Interoperable Security

- Validate your own users
- Validate third party users
- Validate applications that use your service
- Currently unknown methods of authentication

Federated Security

- X509 certificates validate applications
- Claims validate users
  - Authentication generates list of claims
  - Claims are a neutral representation
  - Accept claims from third party identity services
  - Authorize based on claims
  - Use claims today to leverage for future (Geneva)

Claims

```
namespace System.IdentityModel.Claims
{
  public class Claim
   {
    public Claim(string claimType, object resource,
                 string right);
    public string ClaimType { get; }
    public object Resource { get; }
    public string Right { get; }
...
}

Claim c=CreateClaim("AddUsers",affil,Rights.PossessProperty);

List<Claim> claims = new List<Claim>(1);
claims.Add(c);
ClaimSet cs = new DefaultClaimSet(claims);
```

Thread Principal

```
class OurPrincipal : IOurPrincipal, IPrincipal{}

public interface IOurPrincipal
{
    ClaimSet Claims { get; }
    bool HasRequiredClaims(ClaimSet claims);
}

namespace System.Security.Principal
{
    public interface IPrincipal
    {
        IIdentity Identity { get; }
        bool IsInRole(string role);
    }
}
```

Authorization Policy

```
public class ServiceAuthorizationPolicy :
                                       IAuthorizationPolicy
{
    public bool Evaluate(EvaluationContext context, ref
                                              object state)
    {
      ...
      ClaimSet userClaims = LookupUserClaims(user);
      GenericIdentity identity = new GenericIdentity(user);
      IOurPrincipal principal = new
                        OurPrincipal(identity, userClaims);
      context.Properties["Principal"] = principal;
      context.AddClaimSet(this, userClaims);
      ...
     }
     ...
}
```

Where to Authorize?

- Security infrastructure
- Business Logic

Security System Authorization

```
C class AuthorizationManager : ServiceAuthorizationManager
{
  protected override bool CheckAccessCore(OperationContext oc)
 {
    string action =
             oc.RequestContext.RequestMessage.Headers.Action;
    ClaimSet requiredClaims = FindClaimsForAction(action);
    foreach (ClaimSet cs in
      oc.ServiceSecurityContext.AuthorizationContext.ClaimSets)
    {
      foreach (Claim required in requiredClaims)
      {
          bool found = cs.ContainsClaim(required);
          if (found == false)
              return false;
      }
    }
  }
```

Service Authorization

```
OurPrincipal p = Thread.CurrentPrincipal as OurPrincipal;
ClaimSet requiredClaims = GetRequiredClaims(action);
bool result = p.HasRequiredClaims(requiredClaims);
```

Logging is Not Auditing

- Debugging in the cloud requires logging
- Audit based on business requirements
- Business Health Monitoring

Architectural Problems Magnified

- Messages are not remote procedure calls
- Prepare for the future by using claims
- Multiple tenants
- Data customization
- Keep tiers decoupled

Moving to Windows Azure

Azure is the "Middle Way"

- Amazon EC2, VM, no failover, recovery
- Google App Engine, restricted app, failover, recovery
- Azure, cloud platform, metadata, failover, recovery

Cloud Economics

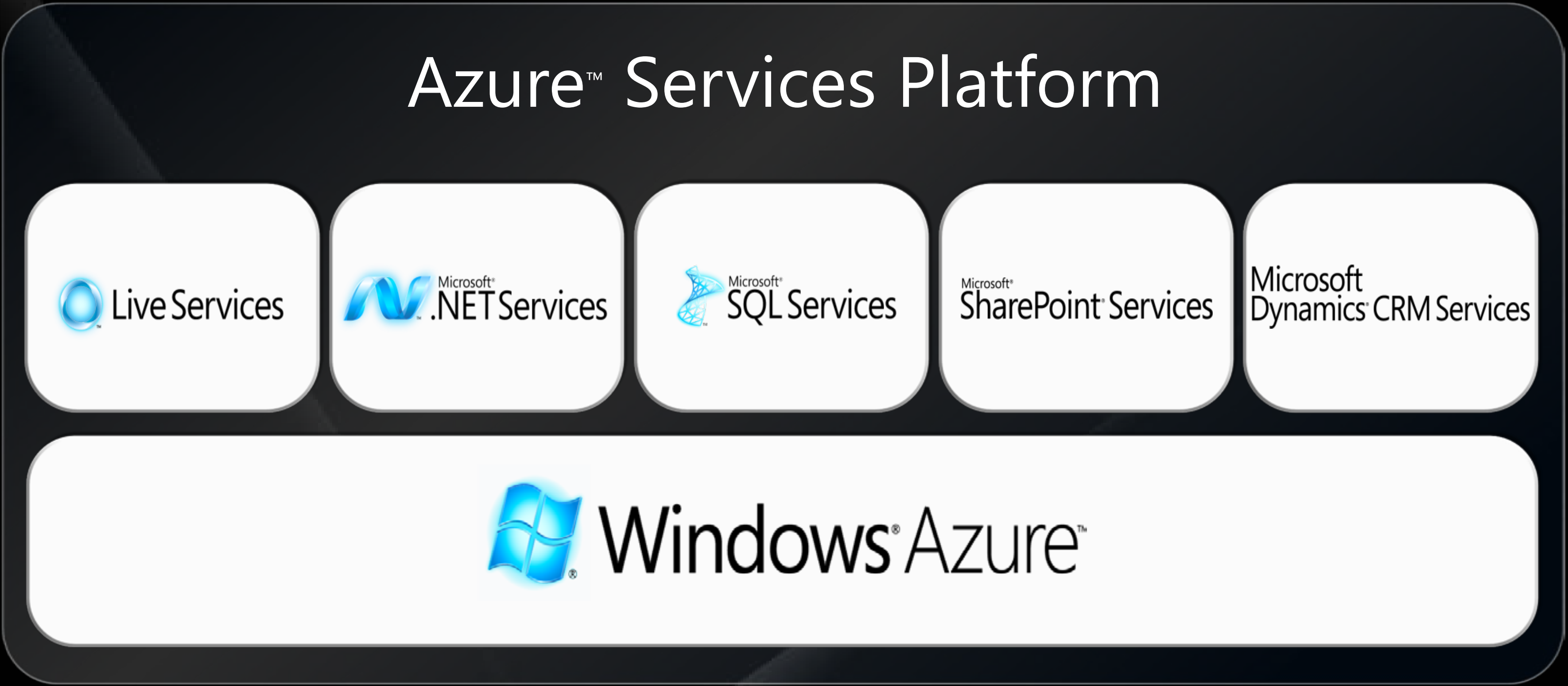
- Economic Calculation
  - Pay as you go
  - Avoid need to build to peak capacity
  - Data available over a wide geographic area
- Risk Sharing
  - Cloud provider must meet peak capacity
  - Cloud provider handles upgrades
- Availability / Service Level Agreement

Moving To Azure

- To move to Azure, think about getting off Azure
- Must Understand Azure application model

Azure Comes In Several Flavors

Windows Live Microsoft Office Live Microsoft Exchange Online Microsoft SharePoint Online Microsoft Dynamics CRM Online



Azure™ Services Platform

Live Services

Microsoft .NET Services

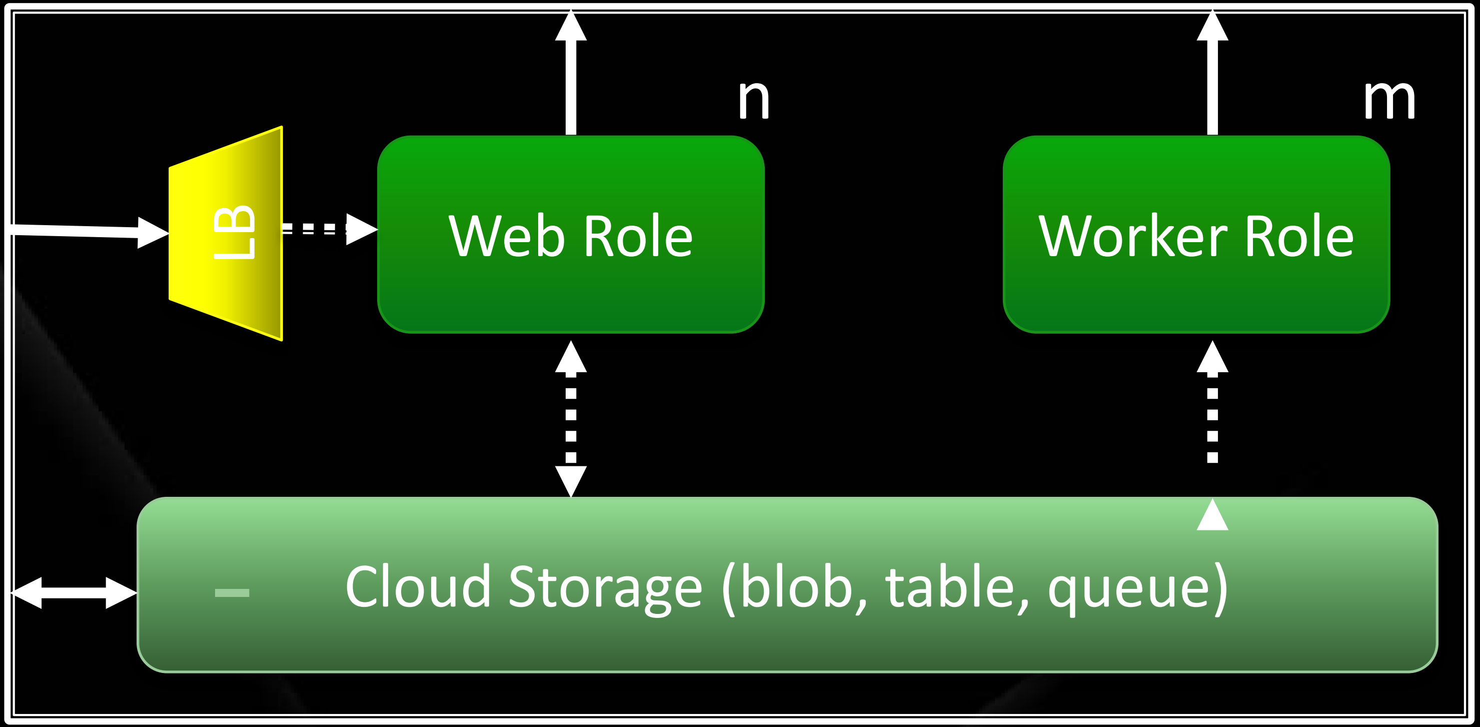
Microsoft SQL Services

Microsoft SharePoint Services

Microsoft Dynamics CRM Services

Windows Azure™

Basic Platform App Architecture



Initial Scenarios

- Look for targets of opportunity
- Cloud services with existing application
- Where is the minimal impedance mismatch?

Scenario: Federated Security

- Access Control .NET Service as STS supplies claims
- Geneva Framework in app to process claims
- Prepare now by using claims for authorization
- Industry standards so easy to replace

Scenario: Hosted SQL Server (SDS)

- From classic or private hosted application.
- Revised to be SQL Server in the sky
  - Tables, Stored Procedures, Triggers, Views, Indices
  - Uses TDS (Tabular Data Stream) Protocol
  - Get Started with SQL Express
- Move back to another SQL Server
- Not Windows Azure Storage Services

Scenario: Move to Azure Platform

- No need to manage infrastructure tier
  - No accounts in the data center
  - No knowledge of which machines app runs on
- Automatic scaling and failover

Mapping Your App to Azure

- Map to pure .NET programming?
  - Web Role is ASP.NET app or Web Service.
  - Worker Role corresponds to Windows Service
- Use Azure platform features, more difficult
  - Blobs, queues, tables

Moving off the Cloud

- Duplicate Google API?
- Amazon is the easiest
- With Azure it depends…
  - .NET Framework with Worker or Web roles
  - Use blobs, queues, tables, need to rewrite data tier

Moving to the Cloud…

- Not an all or nothing process
- Can move parts over time to the cloud

Long Term Process

- Economics are compelling especially for SMB, but so are the architectural challenges
- Usual analogy is to electric power, but data has identity and latency, electrons do not
- Institutional change has to come as well
- People overestimate what can be done in 2 years, but underestimate what happens in 10.

Conclusions

- Architecture is based on a Business Model
- Business Models will drive Cloud adoption
- Architects also think about business.
- Design Concepts you know apply to the Cloud
- Moving to the Cloud is not all or nothing.
- Opportunity for small business and startups.

LEARNING LIVES HERE

question & answer

Resources

Microsoft® tech·ed Online

www.microsoft.com/teched

Sessions On-Demand & Community

Microsoft® | Learning

www.microsoft.com/learning

Microsoft Certification & Training Resources

Microsoft® TechNet

http://microsoft.com/technet

Resources for IT Professionals

msdn®

http://microsoft.com/msdn

Resources for Developers

www.microsoft.com/learning
Microsoft Certification and Training Resources

Related Content

Breakout Sessions

- ARC204 An Overview of the Azure Services Platform
- ARC308 Patterns for Moving to the Cloud
- SIA314 Microsoft Code Name "Geneva" Identity Platform Overview
- DTL404 Case Study: Migrating Existing Client Applications to Windows Azure

Interactive Theater Sessions

- SOA01-INT Architecting Enterprise-Grade Cloud Applications
- ARC01-INT Architecting Your Web Application for the Cloud

Track Resources

Resource 1

Resource 2

Resource 3

Resource 4

Complete an evaluation on CommNet and enter to win!

Microsoft®

Your potential. Our passion.™

The information herein is for informational purposes only and represents the current view of Microsoft Corporation as of the date of this presentation. Because Microsoft must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information provided after the date of this presentation. MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, AS TO THE INFORMATION IN THIS PRESENTATION.