

# Azure by Demo – From 0 to 60



**DEVELOPMENTOR**

DEVELOPING PEOPLE WHO DEVELOP SOFTWARE

# Goal Application – The Blog Engine

## Blog Engine Features

- **Anyone can read entries**
- **Only authenticated users can post entries**
- **Want to be able to upload images**
- **Images should be automatically “thumbnailed”**
- **Blogs can contain any valid html**

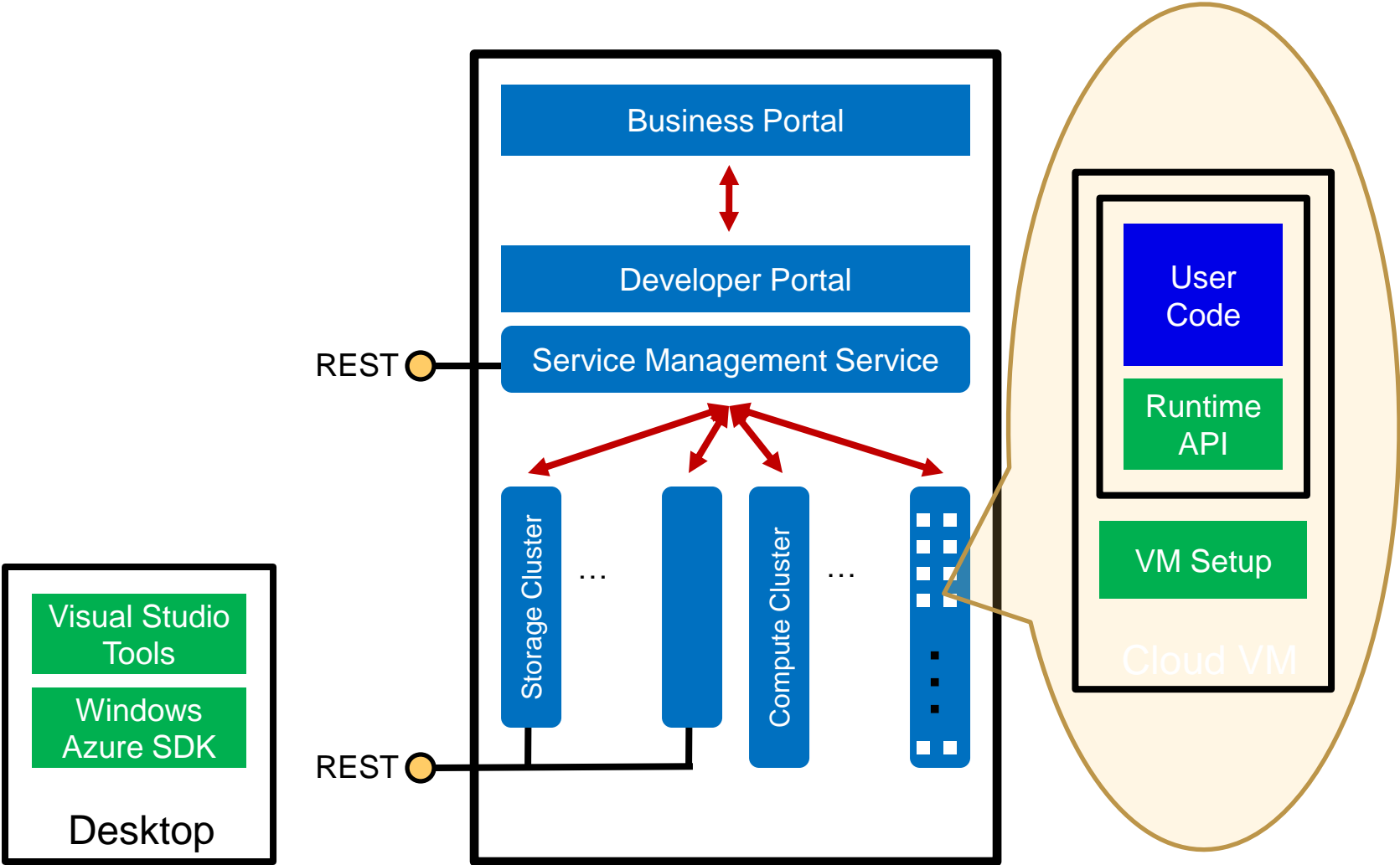


# Azure features (demos) in order

- **Web Roles**
- **Access Control Service**
- **Local Storage**
- **Worker Roles and Inter-role communication**
- **Blobs**
- **Queues**
- **Tables**
- **Deployment**



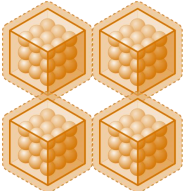
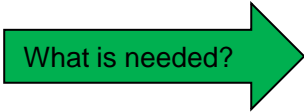
# Windows Azure in a Nutshell



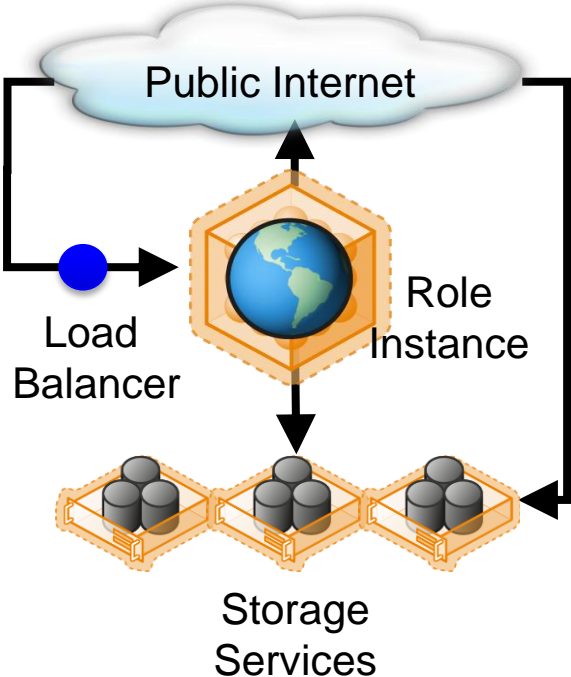
# Service Definition



Service model



Fabric Controller



# Demo #1 – Web Roles

- **Web Role runs under IIS**
- **Either Create new or Add existing**
- **All Roles defined in Service definition**
- **Roles run/debug on Development Fabric**
- **Can have updateable config settings**
- **All Roles have a lifetime** (more on that later)



## Demo #2 – Access Control Service

- **Part of the Azure Platform – AppFabric**
- **Federates with Identity Providers**
- **Can use multiple Relying Parties (dev, test, production)**
- **Use WIF wizard to set it up\***
- **Incoming Identity is a Claims Principal**
- **Integrates with existing .NET security model**

## Demo #3 – Local Storage

- **Should not use it for real storage**
- **Typically used for caching**
- **Pre-allocated by clicking on role properties**





## Demo #4 – Worker Roles and Inter-Role Communication

- **Worker Roles are like NT Services**
- **Role Lifetime – RoleEntrypoint: OnStart, OnStop, Run**
- **Two types of communication endpoints**
  - Input and Internal
- **With either one - listen on a dynamic port**
- **RoleEnvironment is gateway to Service Runtime**
- **Get events/info about other role instances**



## Demo #5 – Blob Storage

- **Storing named files (and metadata)**
- **REST based interface**
- **Blobs stored in containers**
- **Containers have accessibility**
- **Blob name is the partition name**
- **Blob names can be "complex"**
- **Storage client API wrapper**
- **Create a CloudStorageAccount and create a client**
- **Two special types - block and page**
- **Supports for CDNs for higher throughput**



# Block blobs

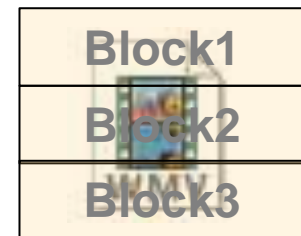
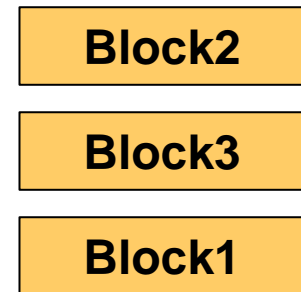
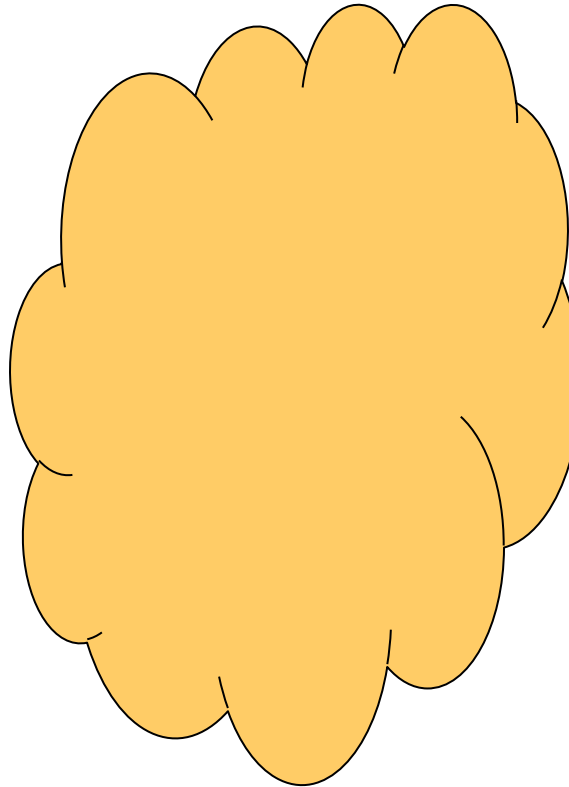
PUT <http://develop.blob.core.windows.net/webcasts/WebCast/Storage.wmv?comp=block&blockid=Block1>



WebCast/Storage.wmv

```
PutBlock("Block1") →  
PutBlock("Block2") →  
PutBlock("Block3") →
```

```
PutBlockList →  
"Block1"  
"Block2"  
"Block3"
```



WebCast/Storage.wmv



## Demo #6 – Queue Storage

- **Designed for role communication**
- **Best effort FIFO**
- **Queues are named**
- **No limit on queue length**
- **Queue name becomes partition name**
- **Messages max 8Kb**
- **Can send pointers or batches**



## Demo #7 – Table Storage

- **No-SQL type structured storage**
- **No relations, schema-free**
- **Stores entities (key-value pairs)**
- **3 required columns (partition key, row key, timestamp)**
- **Partition key ~ clustered index**
- **Partition key and row key = composite key**
- **Queries sans partition key span machines**



## Demo #8 - Deployment

- **Through Web Site**
- **From Visual Studio**
- **Two slots (Staging and production)**
- **Initial deployment can be swapped**
- **Configuration changed without restart**
- **Fault domains and update domains**
- **Automatic and Manual In-place upgrades**

