Cloud Scalability Patterns

Cloud scalability concepts & patterns explained in the context of **Windows Azure Platform** services

GITCA's 24 Hours in the Cloud event gitca.org

19-April-2011

Boston Azure User Group
http://bostonazure.org
@bostonazure

Bill Wilder
http://blog.codingoutloud.com
@codingoutloud

Copyright (c) 2011, Bill Wilder – Use allowed under Creative Commons license
http://creativecommons.org/licenses/by-nc-sa/3.0/

Bill Wilder has been a software professional for over 20 years. In 2009 he founded the **Boston Azure User Group**, an in-person cloud community which gets together monthly to learn about the **Windows Azure platform** through prepared talks and hands-on coding. Bill is a **Windows Azure MVP**, an active speaker, blogger (**blog.codingoutloud.com**), and tweeter (**@codingoutloud**) on technology matters and soft skills for technologists, a member of Boston West **Toastmasters**, and has a day job as a .NET-focused enterprise architect.

BostonAzure.org

Overview of Scalability Topics

1. **What is Scalability?** (10 minutes)
2. **Scaling Data** (20 minutes)
3. **Scaling Compute** (15 minutes)
4. **Q&A** (15 minutes)

Overview of Scalability Topics

1. **What is Scalability?** (10 minutes)
   - Concepts
   - Terminology
2. **Scaling Data** (20 minutes)
3. **Scaling Compute** (15 minutes)
4. **Q&A** (15 minutes)

What does it mean to **Scale**?

- **Scale != Performance**

  Scalable iff Performance constant as it grows

- **Scale the Number of Users**

  Users experience same performance when system has x users as with 10x, 100x, 1000x …

- **Scale the Volume of Data**

  Same performance when system is managing x bytes of data as with 10x, 100x, 1000x …

Options: **Scale Up** and **Scale Out**

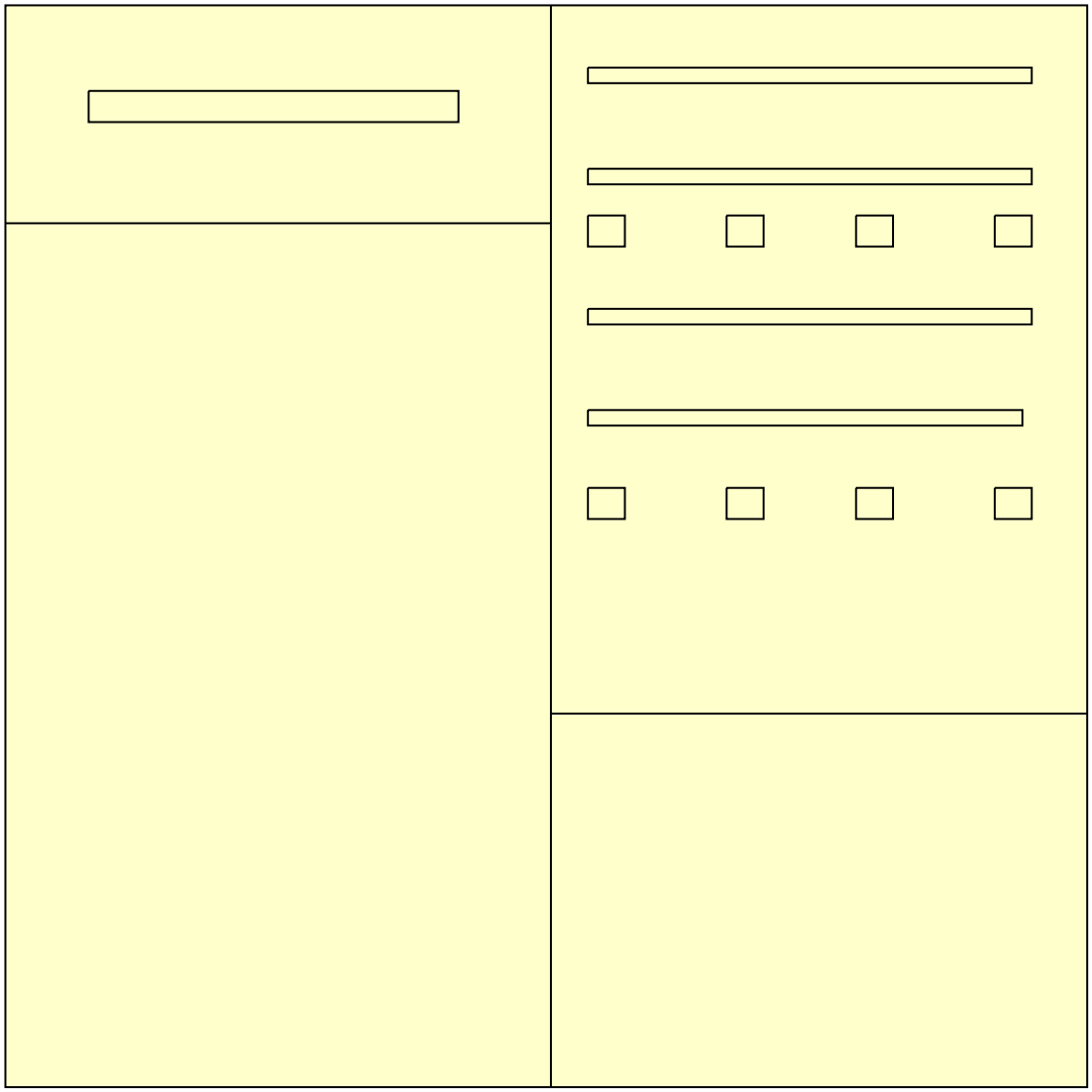
Terminology:

Scaling Up == Vertical Scaling

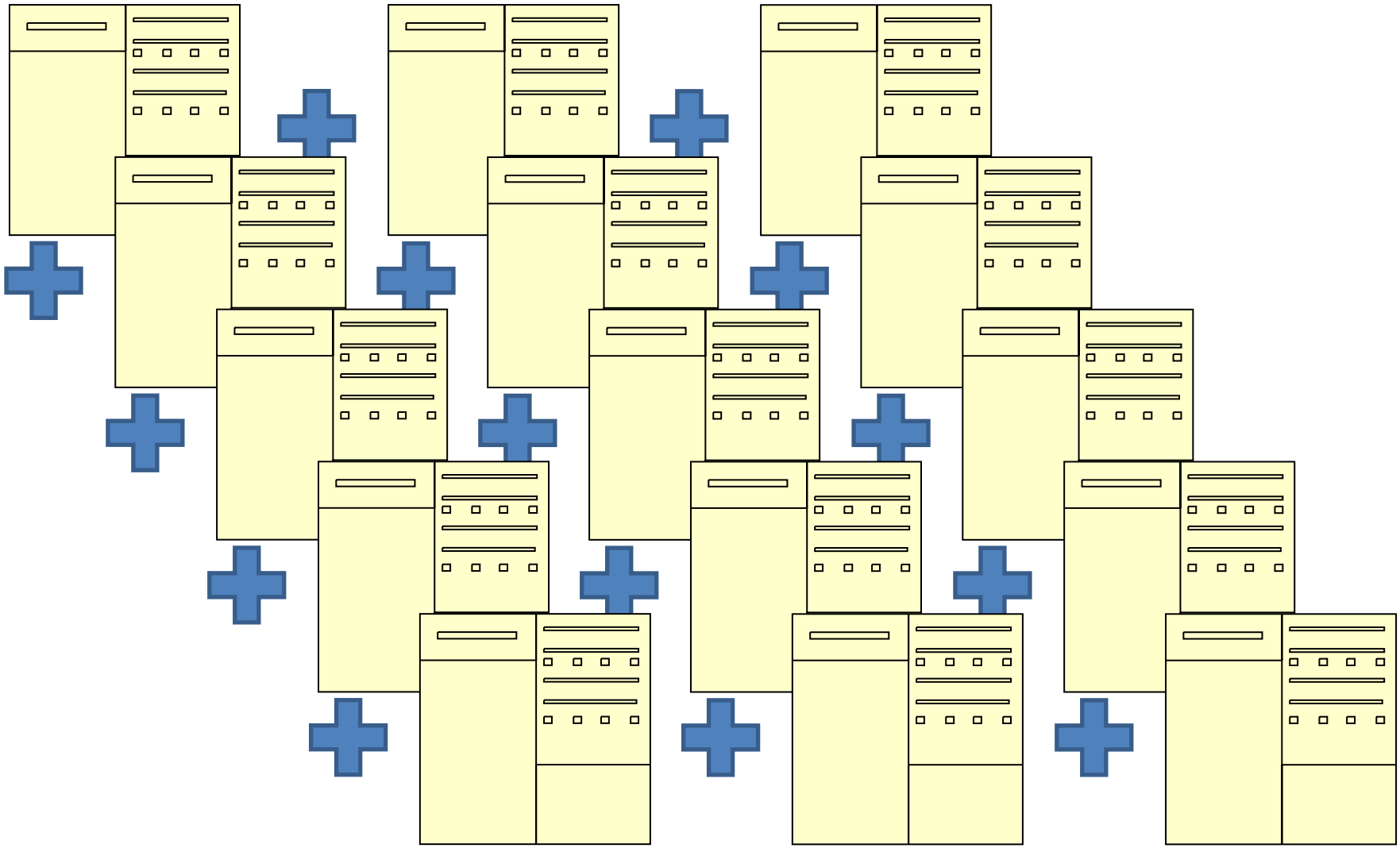
Scaling Out == Horizontal Scaling

- **Architectural Decision**
  - Big decision… hard to change

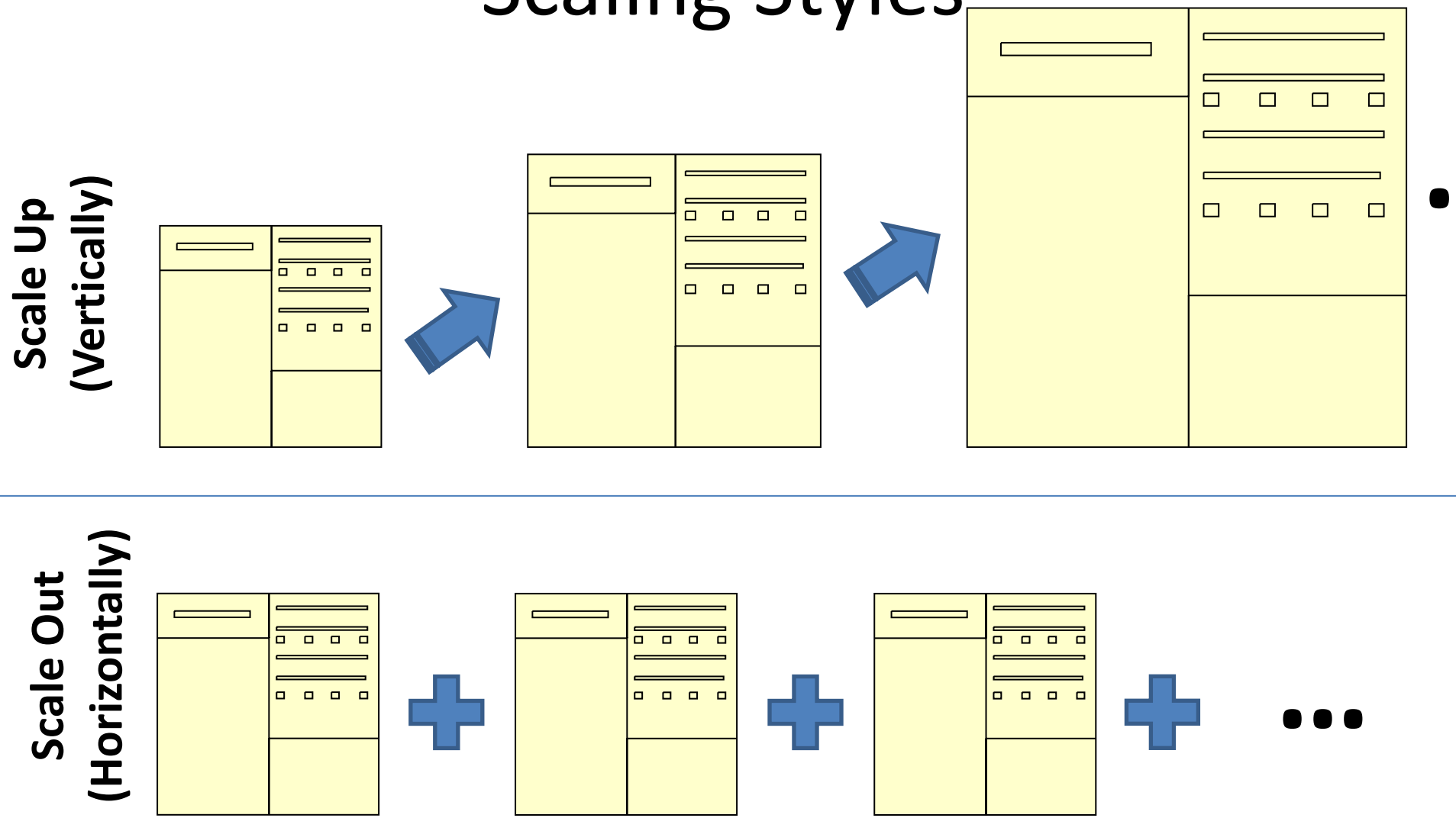
Scaling Up: Scaling the Box



Scaling Out: Adding Boxes



Scaling Styles



Eventual Consistency

- Property of a system such that not all records of state guaranteed to agree at any given point in time.
  - Applicable to whole systems or parts of systems (such as a database)
- As opposed to Strongly Consistent (or Instantly Consistent)
- Eventual Consistency is nature characteristic of a **useful, scalable** distributed systems

Essential Scale Out Patterns

- **Data Scaling Patterns**
  - **Sharding:** Logical database comprised of multiple physical databases, if data too big for single physical db
  - **NoSQL:** “Not Only SQL” – a family of approaches using simplified database model; CAP-friendly
- **Computational Scaling Patterns**
  - **CQRS: C**ommand **Q**uery **R**esponsibility **Segregation**

Overview of Scalability Topics

1. **What is Scalability?** (5 minutes)
2. **Scaling Data** (20 minutes)
   - Sharding with SQL Azure
   - NoSQL with Azure Tables
3. **Scaling Compute** (20 minutes)
4. **Q&A** (15 minutes)

Overview of Scalability Topics

1. **What is Scalability?** (10 minutes)
2. **Scaling Data** (20 minutes)
   - **Sharding with SQL Azure**
   - NoSQL with Azure Tables
3. **Scaling Compute** (15 minutes)
4. **Q&A** (15 minutes)

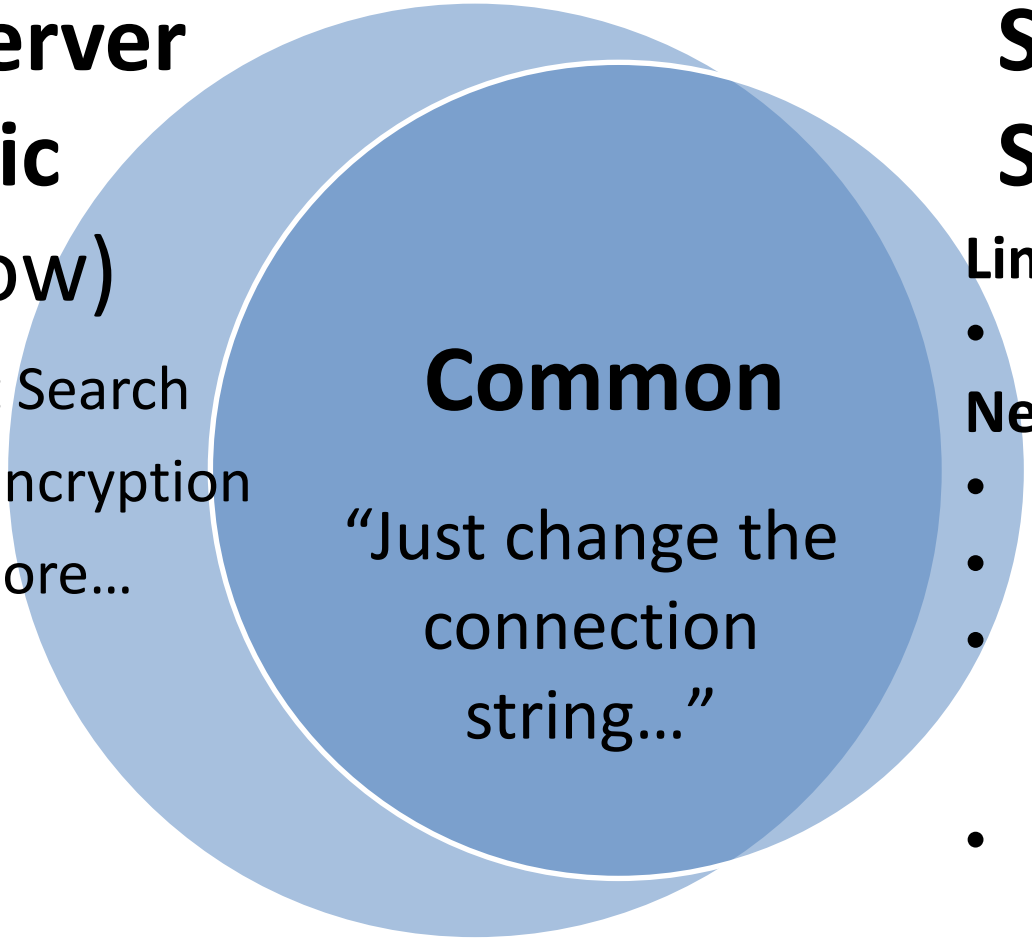
What is Sharding?

- **Problem:** one database can't handle all the data
  - Too big, not performant, needs geo distribution, …
- **Solution:** split data across multiple databases
  - One **Logical Database**, multiple **Physical Databases**
- Each Physical Database Node is a **Shard**
- Most scalable is **Shared Nothing** design
  - May require some denormalization (duplication)

Sharding is Difficult

- What defines a shard? (Where to put stuff?)
  - Example by geography: customer_us, customer_fr, customer_cn, customer_ie, …
  - Use same approach to find records
- What happens if a shard gets too big?
  - Rebalancing shards can get complex
  - Foursquare case study is interesting
- Query / join / transact **across** shards
- Cache coherence, connection pool management

SQL Azure **is** SQL Server Except…



SQL Server Specific (for now)

- Full Text Search
- Native Encryption
- Many more…

Common

"Just change the connection string…"

SQL Azure Specific

Limitations

- 50 GB size limit

New Capabilities

- Highly Available
- Rental model
- Coming: Backups & point-in-time recovery
- **SQL Azure Federations**
- More…

Additional information on Differences:
http://msdn.microsoft.com/en-us/library/ff394115.aspx

SQL Azure Federations for Sharding

- Single “master” database
  - “Query Fanout” makes partitions transparent
  - Instead of **customer_us**, **customer_fr**, etc… we have just **customer** database
- Handles redistributing shards
- Handles cache coherence
- Simplifies connection pooling
- Not a released product offering at this time

- http://blogs.msdn.com/b/cbiyikoglu/archive/2011/01/18/sql-azure-federations-robust-connectivity-model-for-federated-data.aspx

Overview of Scalability Topics

1. **What is Scalability?** (10 minutes)
2. **Scaling Data** (20 minutes)
   - Sharding with SQL Azure
   - **NoSQL with Azure Tables**
3. **Scaling Compute** (15 minutes)
4. **Q&A** (15 minutes)

Persistent Storage Services – Options

Type of Data	Traditional	Azure Way
Relational	SQL Server	SQL Azure
BLOB ("Binary Large Object")	File System, SQL Server	Azure Blobs
File	File System	(Azure Drives) Azure Blobs
Logs	File System, SQL Server, etc.	Azure Blobs Azure Tables
Non-Relational	NoSQL ?	Azure Tables

Not Only SQL

NoSQL Databases

mongoDB, CouchDB: JSON Document Stores

- Amazon Dynamo, Azure Tables: Key Value Stores
  - Dynamo: Eventually Consistent
  - Azure Tables: Strongly Consistent
- Many others!

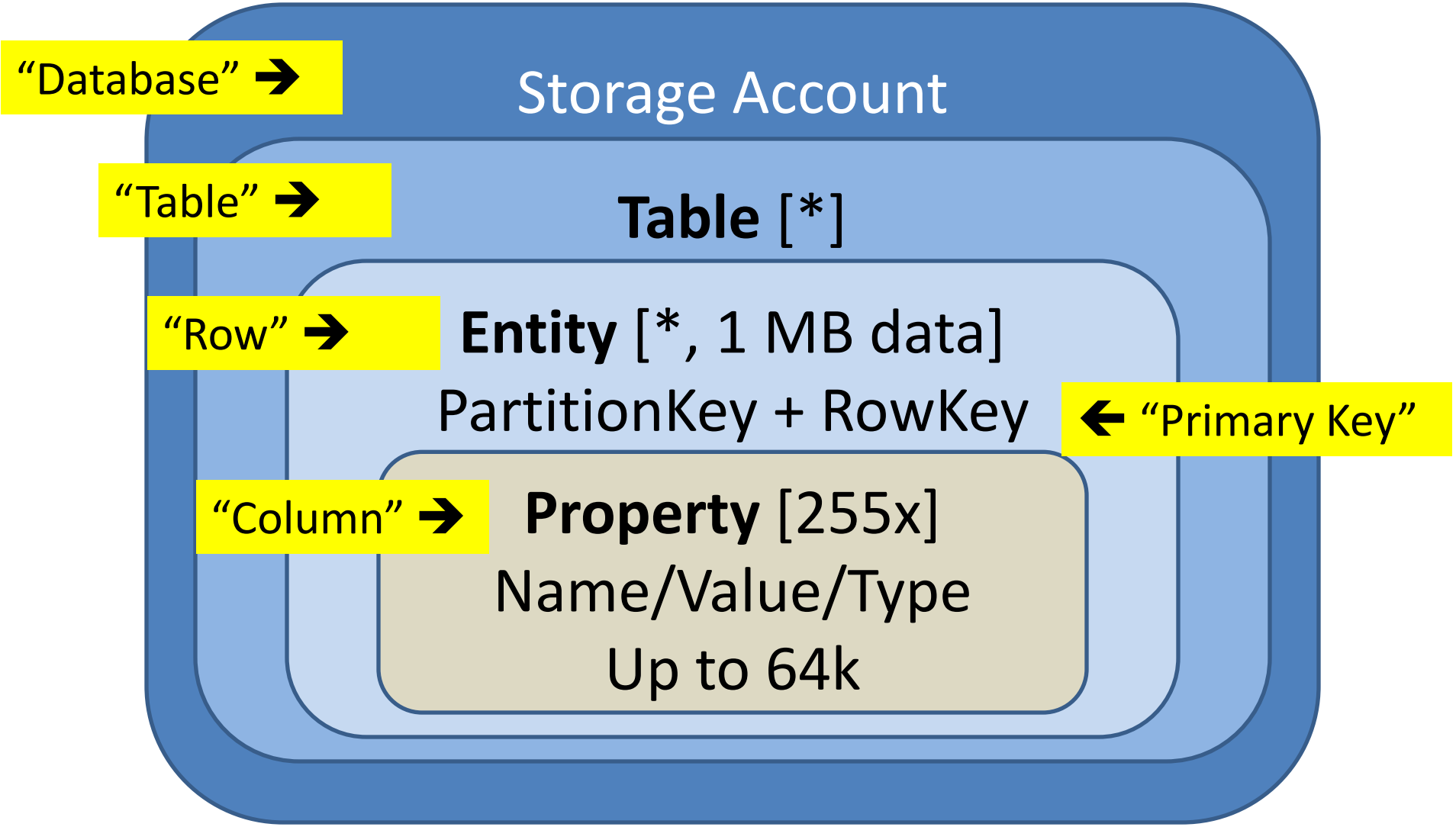
- **Faster, Cheaper**
- **Scales Out**
- **“Simpler”**

- http://en.wikipedia.org/wiki/NoSQL
- http://en.wikipedia.org/wiki/CAP_theorem

SQL Azure (Relational) vs. **Azure Tables** (NoSQL)

Approach	SQL Azure	Azure Tables
Normalization	Normalized	Denormalized
(Duplication)	(No duplication)	(Duplication okay)
Transactions	Distributed	Limited scope
Structure	Schema	Flexible
Responsibility	DBA/Database	Developer/Code
Knobs	Many	Few
Scale	Up (or **Sharding**)	Out
Licensing	License	Rental

Azure Table Storage



Azure Table Storage

- **Best place for granular, semi-structured data**
  - No rigid database schema
  - No complex joins or complex transaction
- Fast and easy to instantiate new containers
  - Strongly Consistent; No performance lag
- Programming model is WCF Data Services
  - All data access and data updates; LINQ
- Durable and optimized to Scale Out
- **No SQL; not managed with SQL Server tooling**

Overview of Scalability Topics

1. **What is Scalability?** (10 minutes)
2. **Scaling Data** (20 minutes)
3. **Scaling Compute** (15 minutes)
   - CQRS pattern with Azure Web Roles, Worker Roles, and Queues
4. **Q&A** (15 minutes)

CQRS Architecture Pattern

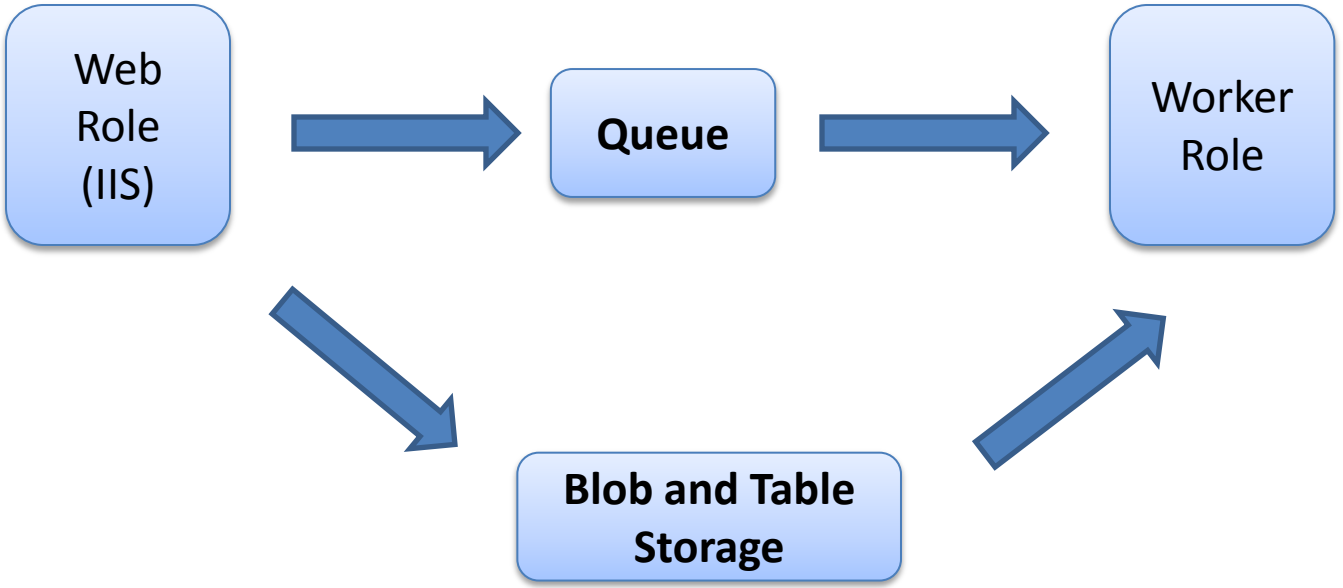
- CQRS = **Command Query Responsibility Segregation**
- Based on notion that actions which Update our system (“Commands”) are a separate architectural concern than those actions which ask for data (“Query”)
- Leads to systems where the Front End (UI) and Backend (Business Logic) are **Loosely Coupled**

CQRS in Windows Azure

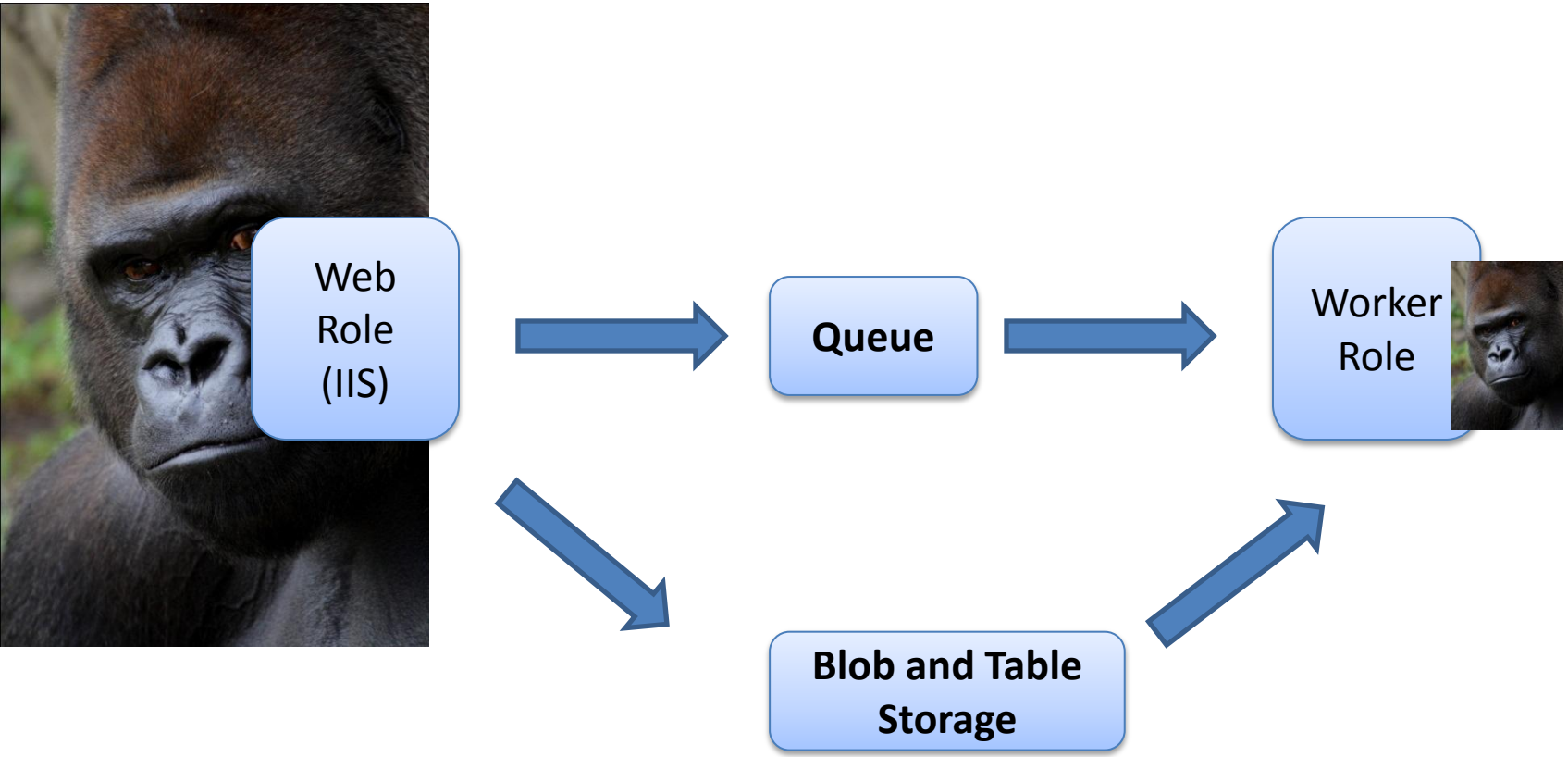
WE NEED:

- Compute resource to run our code
  - ✓**Web Roles** (IIS) and **Worker Roles** (w/o IIS)
- Reliable Queue to communicate
  - ✓Azure Storage **Queues**
- Durable/Persistent Storage
  - ✓Azure Storage **Blobs & Tables**; **SQL Azure**

Key Pattern: Roles + Queues

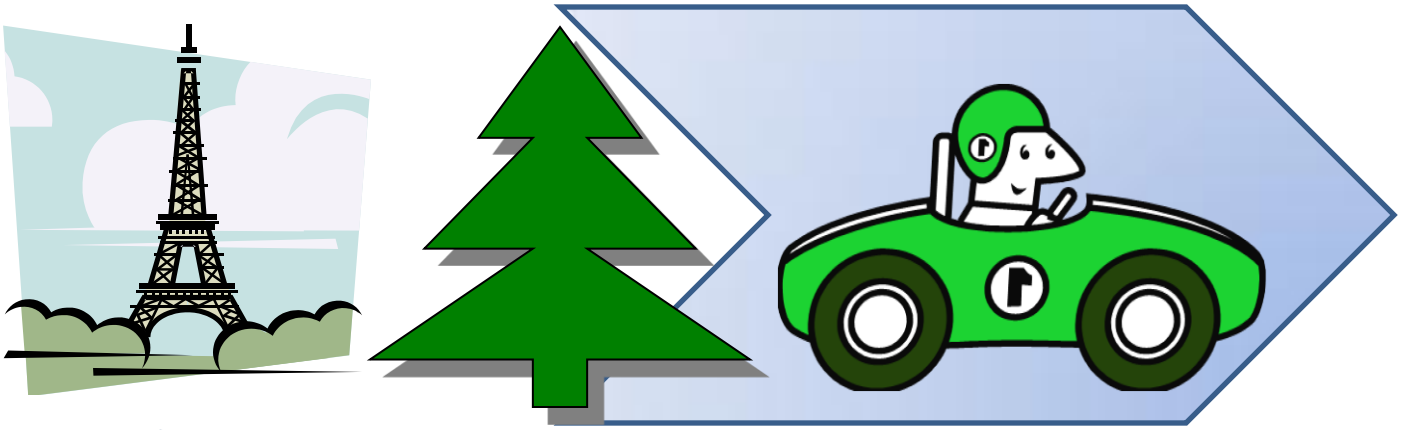


Canonical Example: Thumbnails



Key Point: at first, user does not get the thumbnail (UX implications)

Adding to Queue - ***Conceptual***



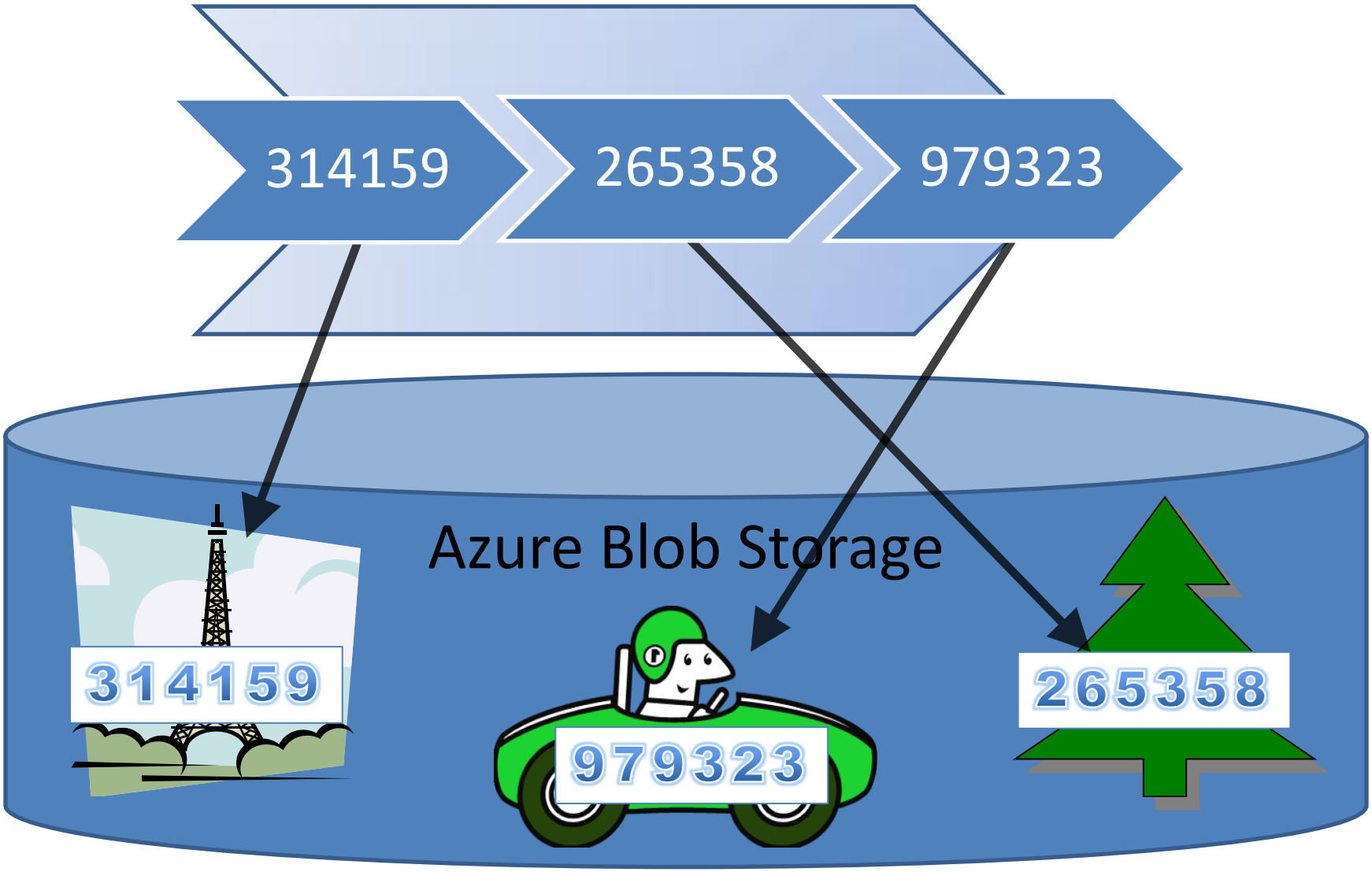
Queue Name: “thumbnailer-7”

http://bostonazure.queue.core.windows.net/thumbnailer-7

Maximum 8 KB per Azure Queue Message

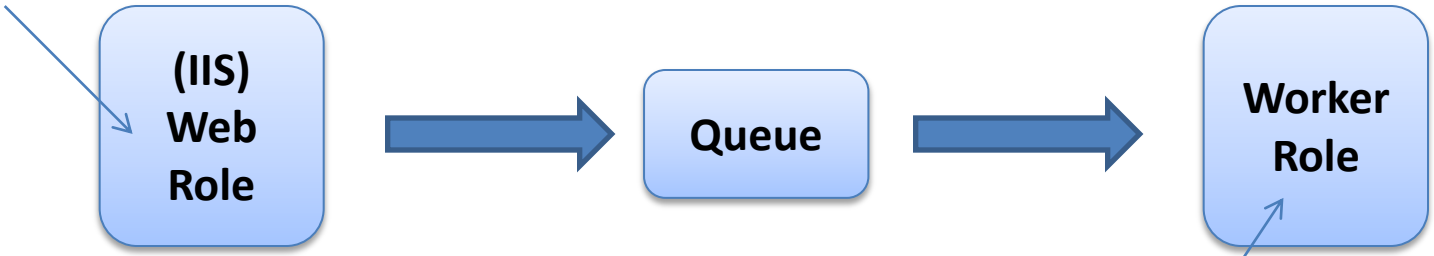
- **Maximum 500 TPS per Azure Queue**
- **Easy, fast to dynamically add new containers**

Adding to Queue - ***Actual***



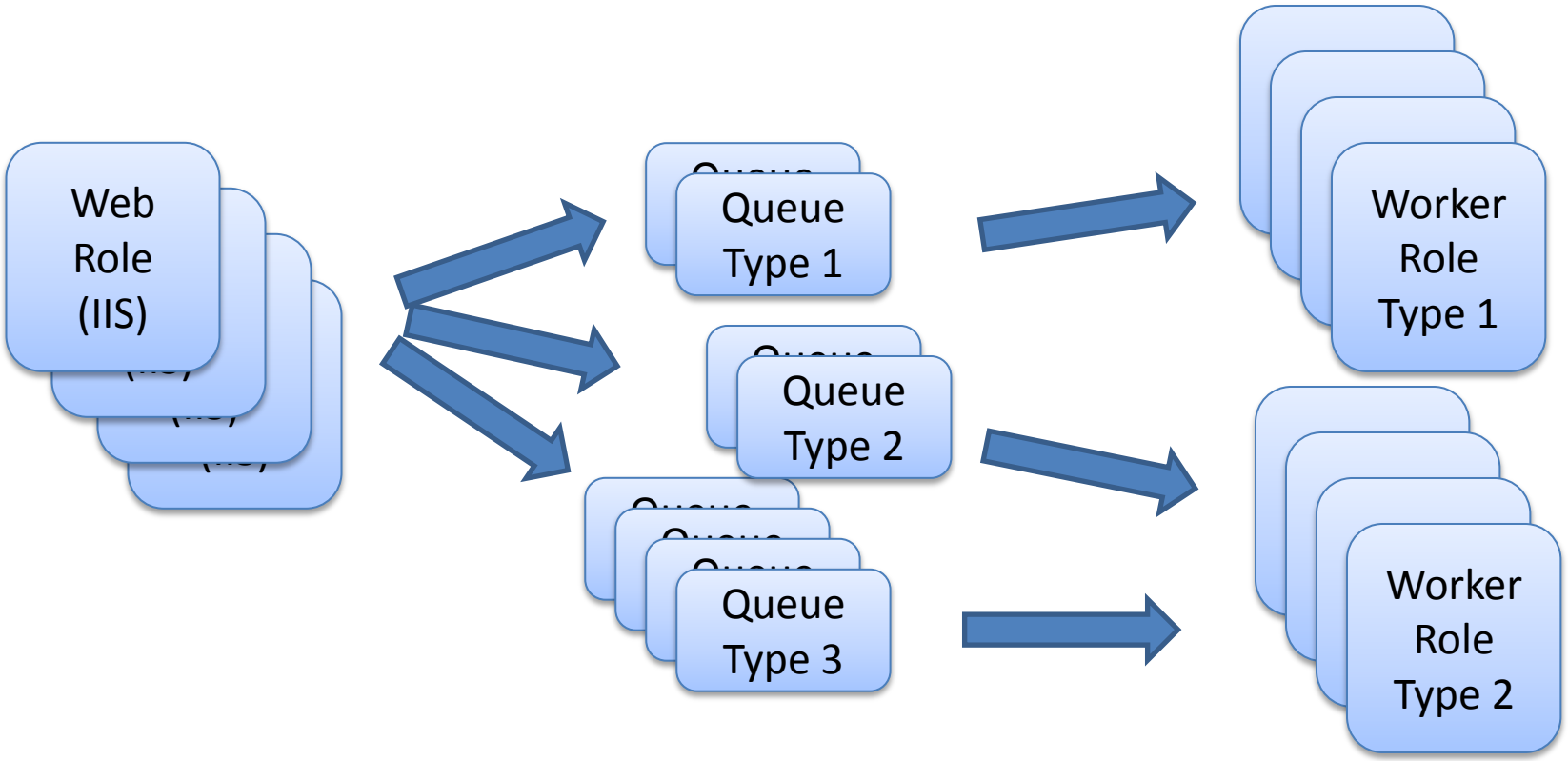
Roles + Queues: API

```
queue.AddMessage(
    new CloudQueueMessage(
        payloadStringOrByteArray));
```




```
CloudQueueMessage
        statusUpdateMessage =
            queue.GetMessage(
                TimeSpan.FromSeconds(10));
... queue.DeleteMessage(statusUpdateMessage);
```

General Case: Many Roles, Many Queues



- **Queue length (and trend) is key data point for tuning Role instance numbers**
- **Easy, fast to dynamically add new queues**

CQRS **requires** Idempotent

- If we do a task twice, end result same as if we did it once
- App-specific concerns dictate approaches
  - Compensating transactions
  - Last in wins
  - Many others possible – hard to say
- Example with Thumnailing

CQRS **expects** Poison Messages

- A Poison Message is not able to be processed
  - Error condition
  - Non-transient reason
  - **CloudQueueMessage.DequeueCount** property
- Strategy One:
  - Fall off the queue (TTL)
  - Message stays in queue for 7 days (default)
- Strategy Two:
  - Specify retry threshold
  - Remove poison messages

CQRS **enables** Responsive

- Response to interactive users is as fast as a work request can be persisted
- Time consuming work done off-line
- Same total resource consumption, better subjective experience
- **UX challenge** – how to express Async to users?
  - Communicate Progress
  - Display Final results

CQRS **enables** Scalable

- Loosely coupled, concern-independent scaling
- Blocking is Bane of Scalability
  - Decoupled front/back ends insulate from other system issues if…
  - Twitter down
  - Email server unreachable
  - Order processing partner doing maintenance
  - Internet connectivity interruption

CQRS **enables** Resilient

- And **Requires** that you “Plan for failure”
- **There will be role restarts**
- Bake in handling of restarts
  - Not an exception case! Expect it!
  - Restarts are routine, system “just keeps working”
- ***If you follow the pattern, the payoff is substantial…***

What's Up?

Aspirin-free Reliability as *EMERGENT PROPERTY*

	Typical Site	Any 1 Role Inst	Overall System
Operating System Upgrade	✗ ?	✗	✓
Application Update / Deploy	✗ ?	✗	✓
Change Topology	✗	✗	✓
Hardware Failure	✗	✗	✓
Software Bug / Crash / Failure	✗	✗	✓
Security Patch	✗ ?	✗	✓

Overview of Scalability Topics

1. **What is Scalability?** (10 minutes)
2. **Scaling Data** (20 minutes)
3. **Scaling Compute** (15 minutes)
4. **Q&A (15 minutes)**
   - Questions?
   - Comments?

Questions?
Comments?
More information?

Free 30-Day Windows Azure & SQL Azure Pass

(in countries where Azure is offered)

- Visit http://bit.ly/BillOnAzure
- Use this promo code: **BillOnAzure**
- You will be provisioned an Azure account valid for 30 days that includes for **FREE**:
  - Three small compute instances
  - Two 1 GB SQL Azure Databases
  - 3 GB of Windows Azure Storage
  - And more…

Windows Azure
Microsoft's Cloud Computing Platform

All Microsoft Sites

Search Windows Azure

Take Bill's advice: try the cloud for 30 days, for free!

Here is what you get (& there's no credit card needed):

Windows Azure
3 Small Compute Instances; 3 GB of Storage; 250,000 Storage Transactions

SQL Azure
2 One-GB Web Edition Databases

AppFabric
100,000 Access Control Transactions; 2 Service Bus Connections

Data Transfers
3 GB In; 3 GB Out

1 SIGN UP FOR YOUR PASS

To begin, select your country and enter a promo code. Don't have a promo code?

United States

BillOnAzure

Submit

*No credit card required

2 INSTALL WINDOWS AZURE TOOLS FOR MICROSOFT VISUAL STUDIO 1.3
Get Windows Azure Tools for Microsoft Visual Studio to start building and debugging applications for Windows Azure.
Get Tools & SDK

3 CREATE YOUR FIRST LOCAL APPLICATION
Learn how to create a simple ASP.NET application in Microsoft Visual Studio for Windows Azure.
Start the tutorial

4 DEPLOY YOUR APPLICATION TO THE CLOUD
Learn how to deploy and run your sample application in Windows Azure.
Start the tutorial

5 EXPLORE ADDITIONAL WINDOWS AZURE PLATFORM OFFERS
Find the Windows Azure platform plan that works best for you.
Learn more

BostonAzure.org

- Boston Azure cloud user group
  - Focused on Microsoft's cloud platform
- Last Thursday, monthly, 6:00-8:30 PM at NERD
  - Food; wifi; free; great topics; growing community
- Follow on Twitter: **@bostonazure**
- More info or to join our email list:

http://www.bostonazure.org

Contact Me

I may be able to speak at your community event

(more likely if you are in vicinity of Boston!)

Bill Wilder

@codingoutloud

http://blog.codingoutloud.com

codingoutloud@gmail.com