Challenges of quality management in cloud applications

Mgr. David Gešvindr
MCSE: Data Platform | MCT | MSP
gesvindr@mail.muni.cz

Motto

- **Classical web application designed for on-premise environment cannot utilize the full potential of the cloud**

- Higher operation costs
- Cloud application has to be designed in different way applying different tactics and patterns
  - Cloud platform offers a wide portfolio of services
  - Anytime anything can fail
  - Need to optimize based on multiple conflicting criteria:
    - High operation costs for relational database x Lower development costs

Outline

1. Introduction to a cloud environment and its foundations
2. Identification of relevant software quality attributes in the cloud
3. Frequent mistakes in cloud application design
4. Cloud specific architectural tactics and guidelines for their application

Outline

1. **Introduction to a cloud environment and its foundations**
2. Identification of relevant software quality attributes in the cloud
3. Frequent mistakes in cloud application design
4. Cloud specific architectural tactics and guidelines for their application

Cloud definition

„Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction.“

- National Institute of Standards and Technology

Characteristics of the cloud

- On-demand self service
- Broad network access
- Resource pooling
- Rapid elasticity
- Measured service

Service model

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

Deployment model

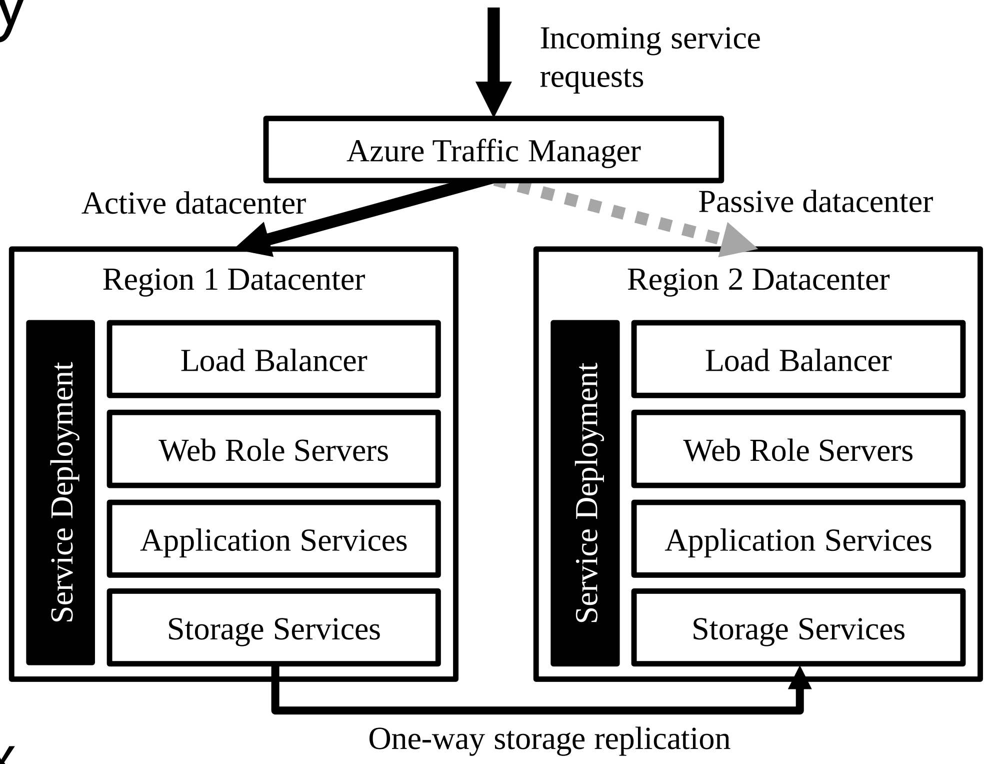
- Public cloud
- Private cloud
- Hybrid cloud

Outline

1. Introduction to a cloud environment and its foundations
2. **Identification of relevant software quality attributes in the cloud**
3. Frequent mistakes in cloud application design
4. Cloud specific architectural tactics and guidelines for their application

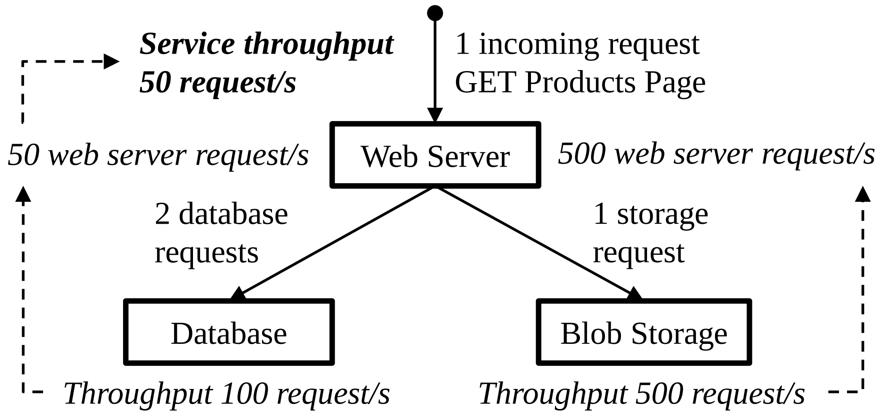
Availability

- Cloud platform offers service availability **99.95-99.99%**
- Be aware of transient errors
  - Need to implement detection and retry policy to prevent random faults
- User error – damaged data, recovery
- Data center outage
  - Design of a cross data center cloud application requires application of complex tactics to work properly



Throughput

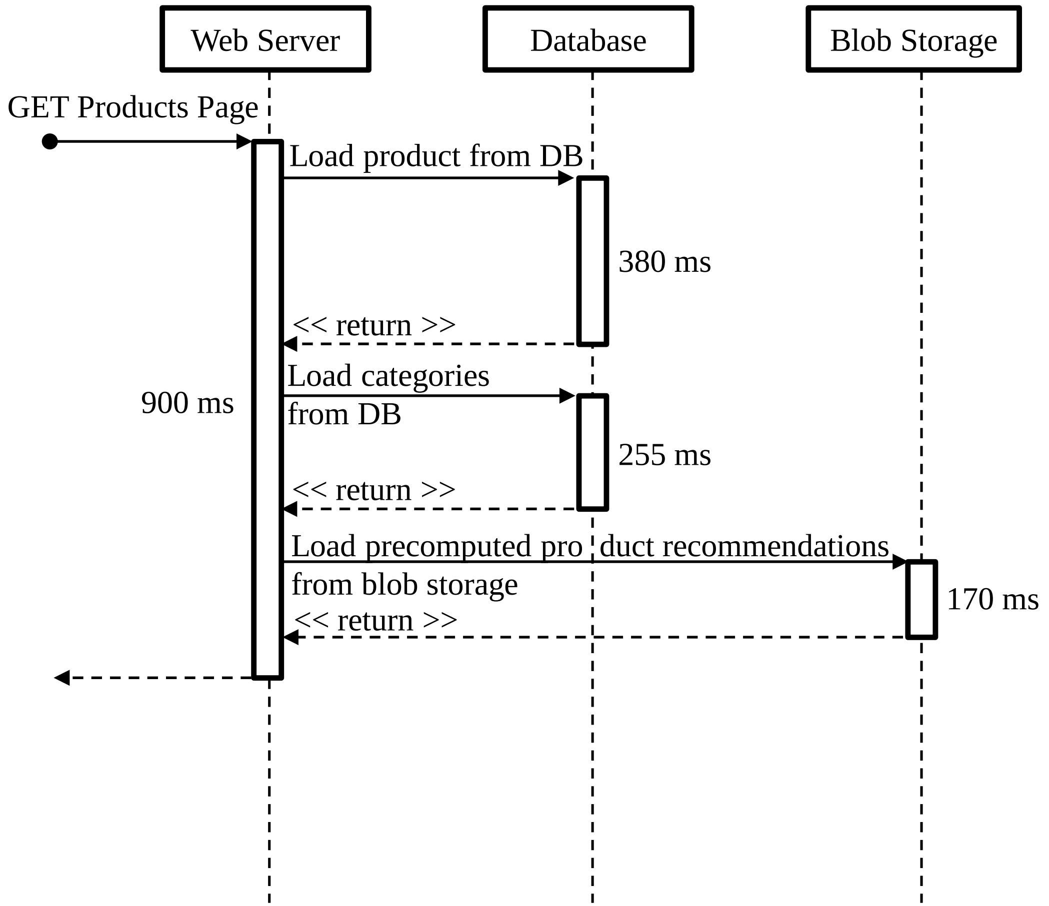
- **Measure of the amount of work an application must perform in a unit of time**
- Throughput is being quantified with a number of transactions, operations or requests that the system can handle per second or other time unit.



- Strongly dependent on throughput of application components involved in the request processing
- Early identification of the bottleneck
- Be aware of the difference between average and peek throughput

Response Time

- Response time is a measure of the latency an application exhibits in processing a business transaction
- Is determined by
  - Communication latency
  - Request processing time



Scalability

- Scalability characterizes how well a solution to some problem will work when the size of the problem increases
- Be aware of the difference:
  - **Performance related attributes** - Specify application behavior for a **static** instance of **cloud environment conguration**
  - **Elasticity** – The degree to which a system is able to adapt to workload changes by provisioning and deprovisioning resources in an autonomic manner
- **If the application is not scalable** it cannot effectively utilize the potentially unlimited amount of processing resources that the cloud platform offers

Operation costs

- **It is necessary to precisely evaluate all operation costs**

- Problem:
  - Service is billed based on real resource usage – how to effectively predict operation costs?

Development costs

- Multiple services offer similar services but the difference is critically important
  - Storage services differs in offered functionality (SQL vs. NoSQL database)
  - Differences in scalability

- Integration costs are based on the functionality, available libraries and tools

- For instance:
  - Azure Storage x Azure SQL Database x DocumentDB

Outline

1. Introduction to a cloud environment and its foundations
2. Identification of relevant software quality attributes in the cloud
3. **Frequent mistakes in cloud application design**
4. Cloud specific architectural tactics and guidelines for their application

Case study: Elections in USA

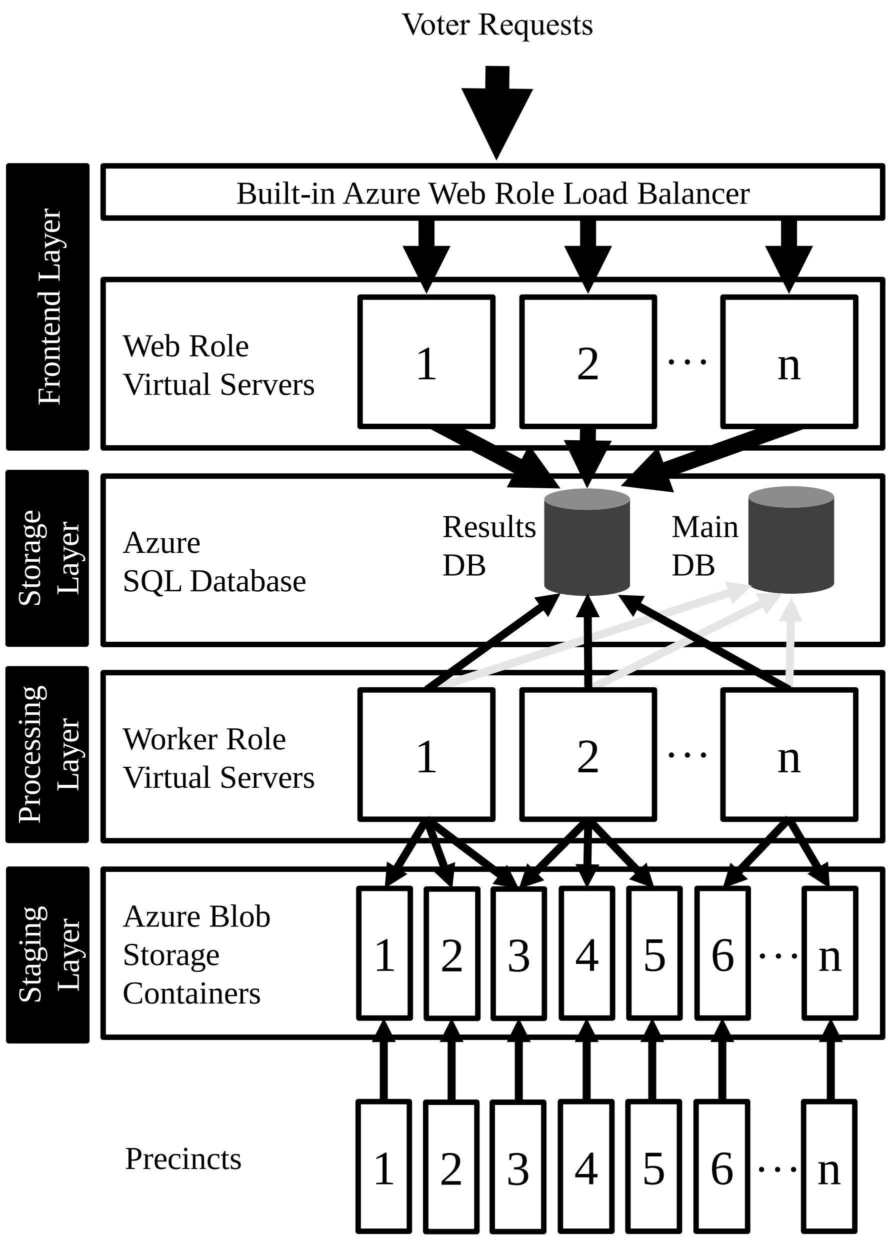
- This case study was presented at TechEd 2014 by Azure CTO **Mark Russinovich**

- Video recording of the session:
  http://channel9.msdn.com/Events/TechEd/Europe/2014/CDP-B337

- **System for presenting results of US elections**

Service architecture

- Election results are uploaded to Azure Storage
- Worker role continuously processes the results
- Processed results are stored in relational database
- Web servers load results from DB



Expected load

- Every user request results **in 10 database requests**
- Expected service load

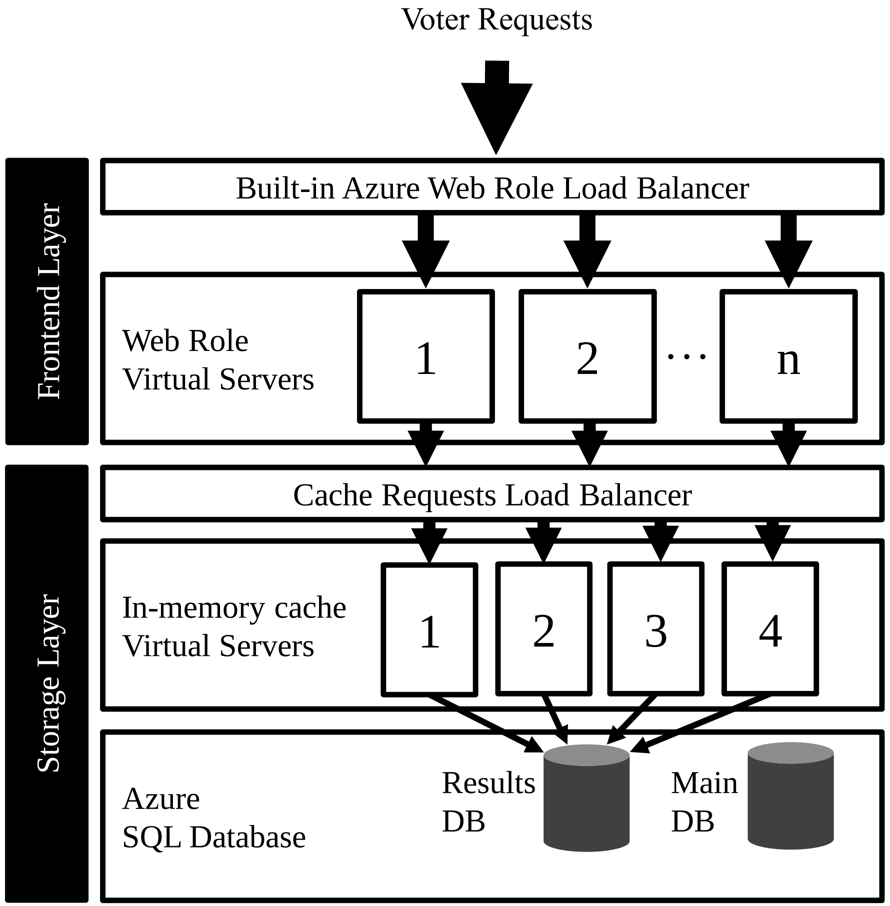
Expected Load				
Scenarios	Expected Page Views	Time Window (hrs)	Page View/sec	10X/pvs DB Calls/sec
Average	10,000,000	4	694	6,944
Peak Hour	6,000,000	1	1,667	16,667

- Problems:
  - Azure SQL throughput is limited up to **1000 requests per second**

Update of application's architecture

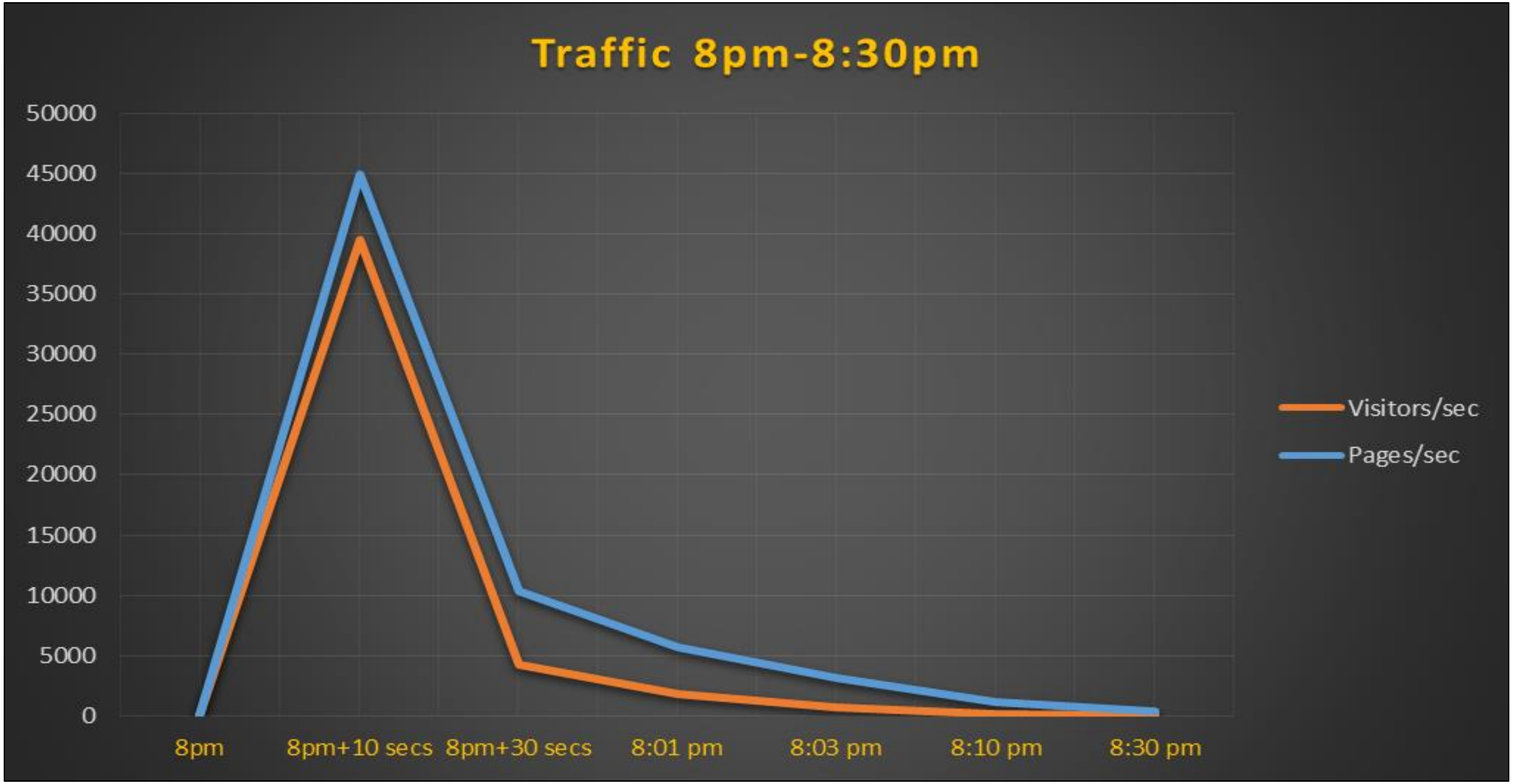
1. Addition of cache layer which is composed of **worker servers** hosting in-memory cache

- **Throughput of the storage increased in order of magnitude**



Rest of the architecture remains the same

Real load



Traffic 8pm-8:30pm
50000
45000
40000
35000
30000
25000
20000
15000
10000
5000
0
8pm
8pm+10 secs
8pm+30 secs
8:01 pm
8:03 pm
8:10 pm
8:30 pm
Visitors/sec
Pages/sec

Allocated capacity

- With database

Time	Actual Page Views	Time Window (sec)	Page View/sec	Calls/sec	Difference Calls/sec	Requests served
8pm+10 secs	448932	10	44893	448932	-447932	0,22%
8pm+30 secs	206925	20	10346	103463	-102463	0,97%
8:01 odp.	171231	30	5708	57077	-56077	1,75%
8:03 odp.	37835	120	3153	31529	-30529	3,17%
8:10 odp.	494423	420	1177	11772	-10772	8,49%
8:30 odp.	416379	1200	347	3470	-2470	28,82%

- With in-memory cache

Time	Actual Page Views	Time Window (sec)	Page View/sec	Calls/sec	Difference Calls/sec	Requests served
8pm+10 secs	448932	10	44893	448932	-288932	35,64%
8pm+30 secs	206925	20	10346	103463	56537	100,00%
8:01 odp.	171231	30	5708	57077	102923	100,00%
8:03 odp.	37835	120	3153	31529	128471	100,00%
8:10 odp.	494423	420	1177	11772	148228	100,00%
8:30 odp.	416379	1200	347	3470	156530	100,00%

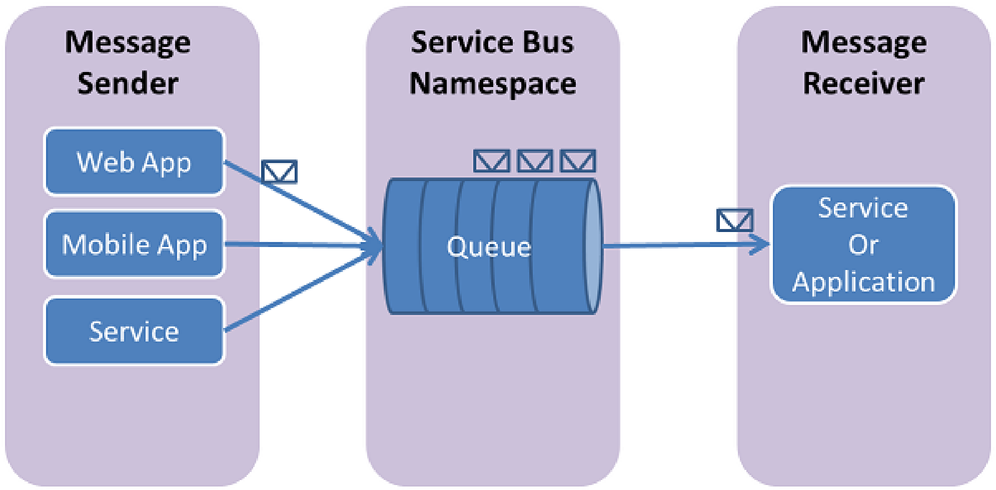
Case study 2: Shopping in Amazonu

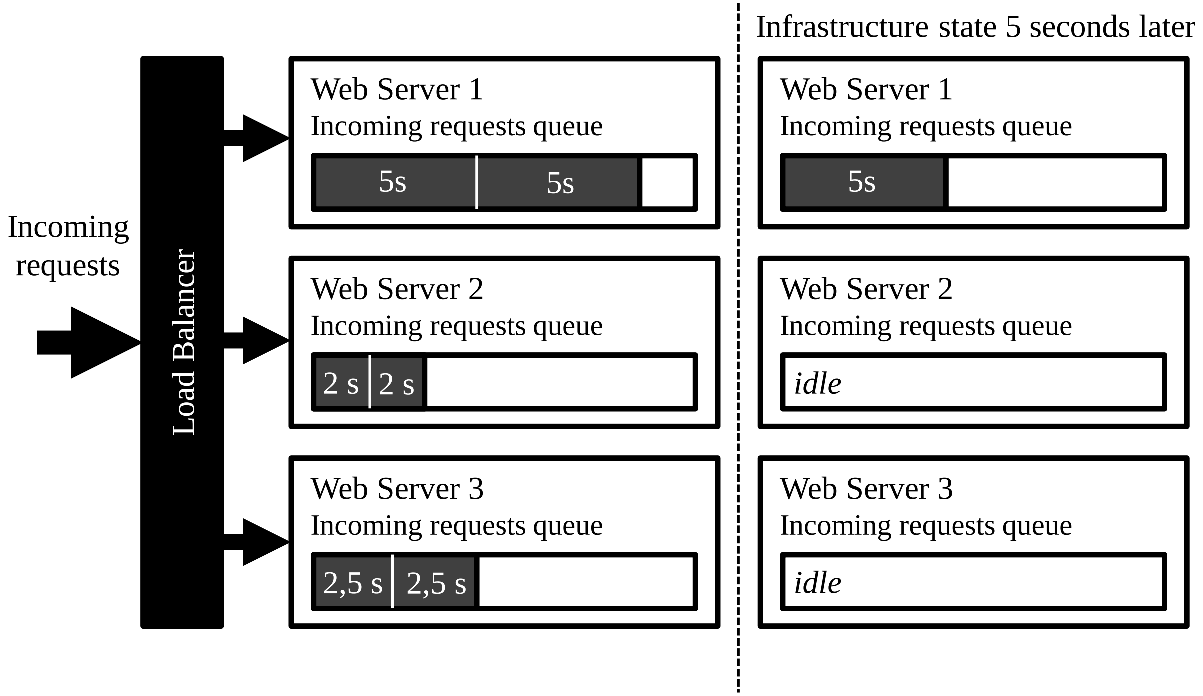
- **Do you know how is rendered a product page in Amazon e-shop?**

- Product page including recommended products for a specific user is pre-rendered as a fragment and stored in S3 storage
- Page transmitted to the user is just simple composition of pre-rendered fragments

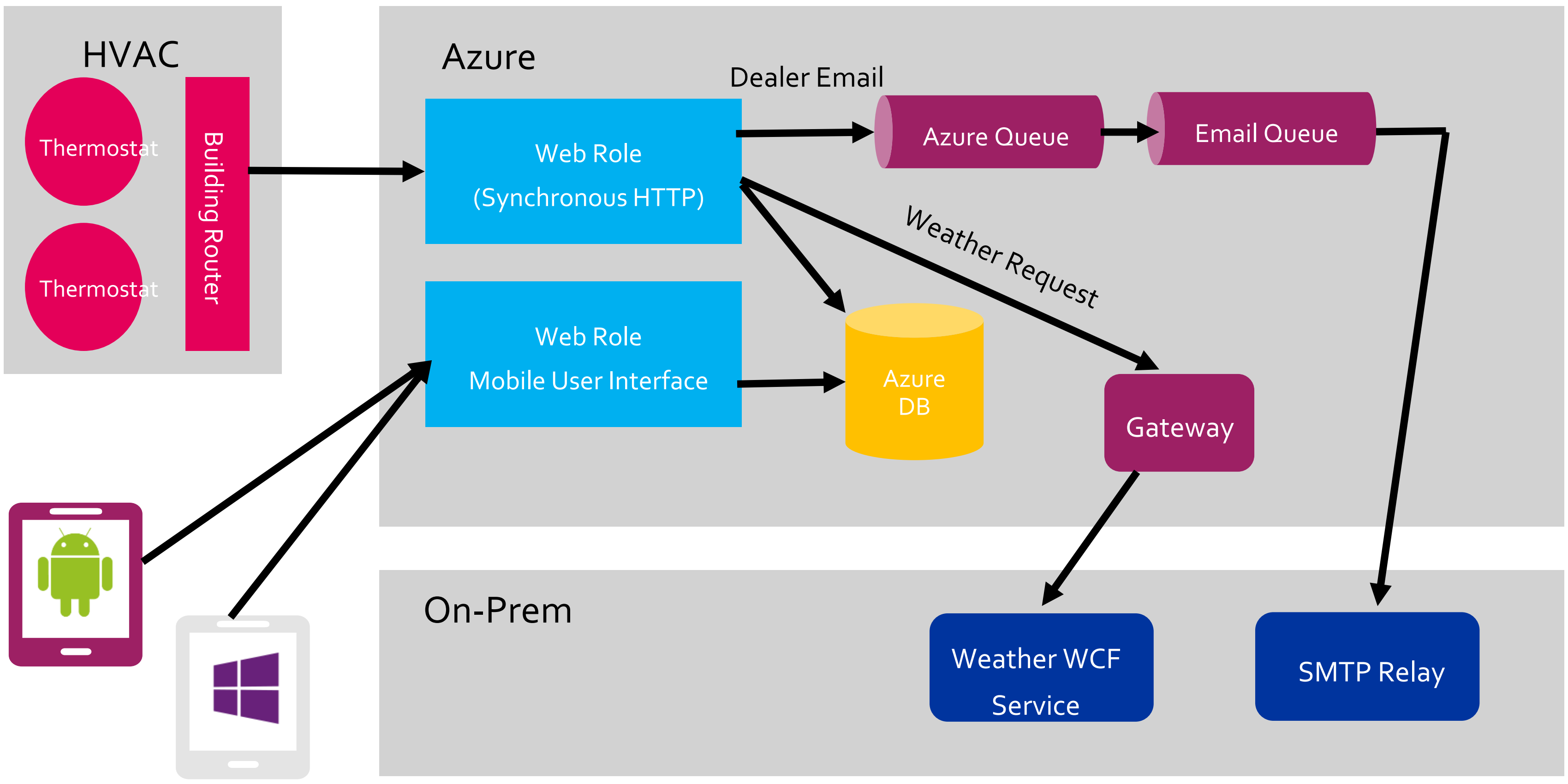
Indirect dependencies

- Compute operations are done asynchronously/independently on synchronous processing of user request
- Compute operation is stored in scalable queue service and worker processes load task definitions from the queue

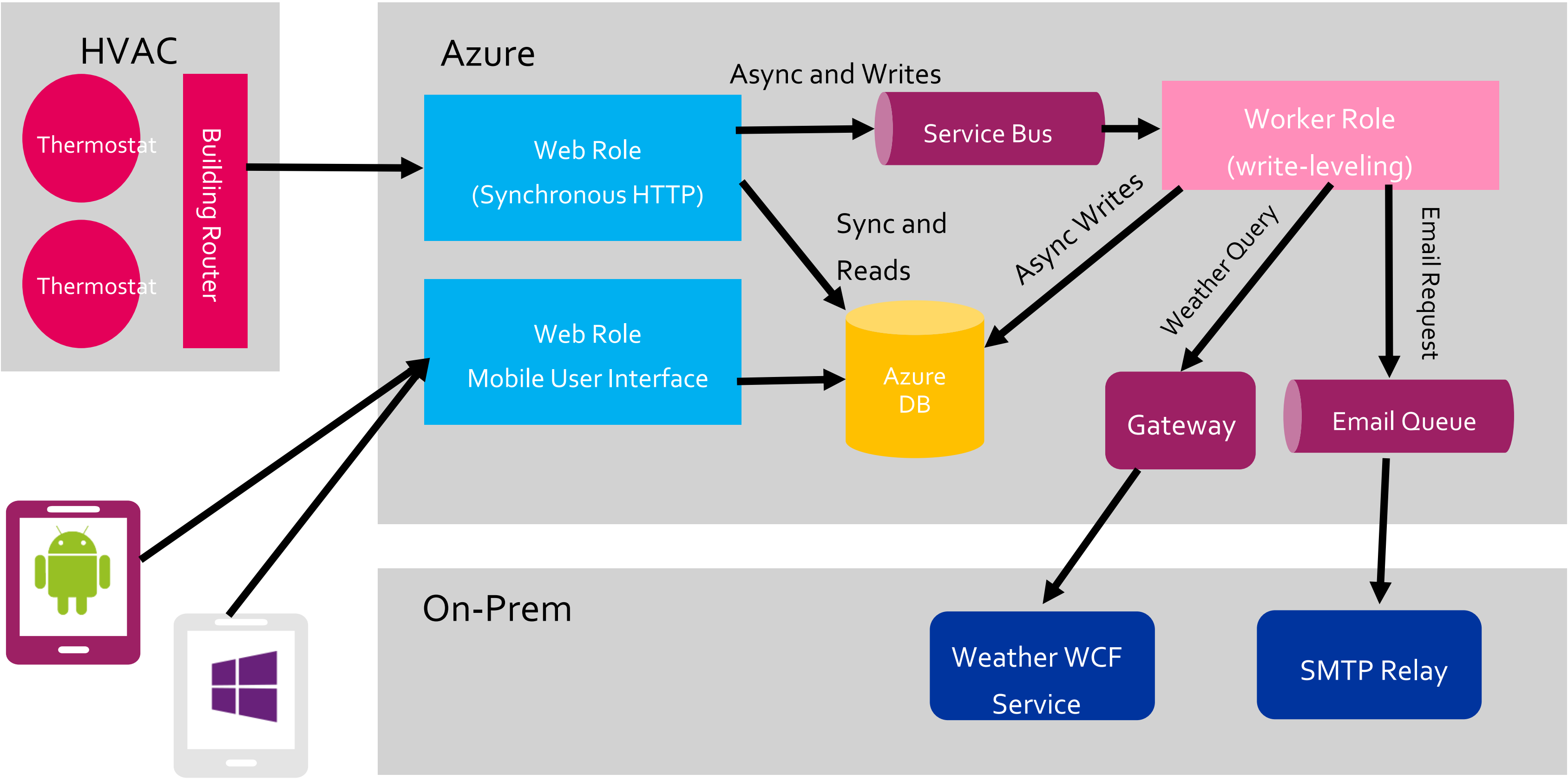




Case study 3: Smart thermostats

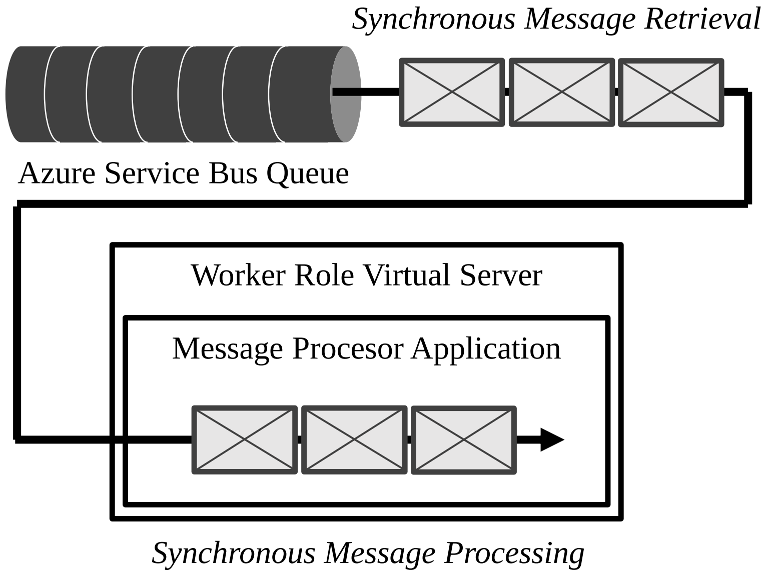


Addition of asynchronous dependencies

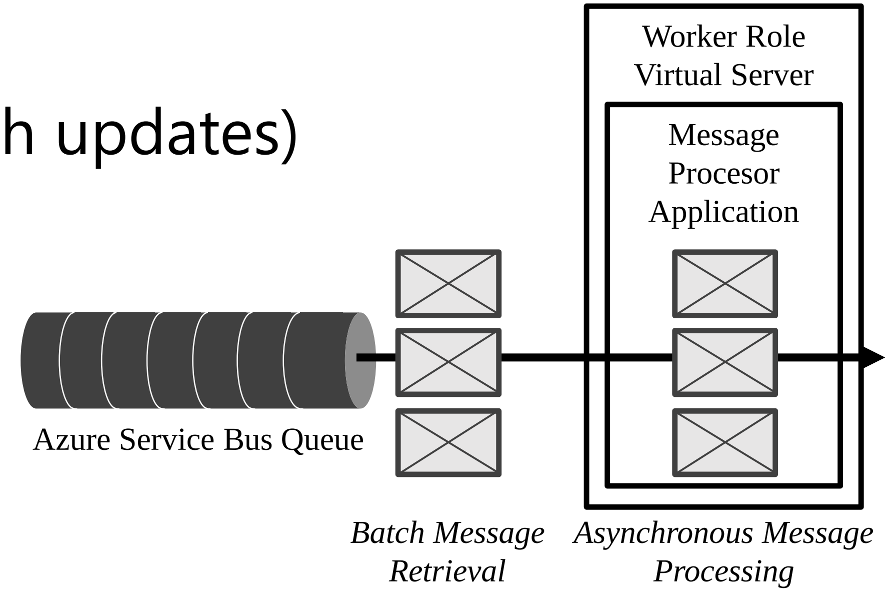


Case study 3: Conclusion

- Initial tests failed with 35 000 connected thermostats
- Goal was 100 000 (150 000) thermostats



- Main issues:
  - Synchronous HTTP handler
  - Row-level updates of the DB (instead of batch updates)
  - Database tuning
  - Queue scalability issues, resolved by an application of partitioning



Outline

1. Introduction to a cloud environment and its foundations
2. Identification of relevant software quality attributes in the cloud
3. Frequent mistakes in cloud application design
4. **Cloud specific architectural tactics and guidelines for their application**

Cloud specific architectural tactics

1. Multi-tiered Storage Tactic
2. Indirect Dependency Tactic
3. Results Pre-computation Tactic
4. Component Co-deployment Tactic

1. Multi-tiered Storage Tactic

- **Goal:** Combine various storage services to enhance their advantages and mitigate their weaknesses

- **Relational Database**
  + Transaction processing, integrity constraints, complex quering
  – Limited scalability, expensive operation
- **NoSQL databáze** (Azure Table Storage)
  + Good scalability, cheap operation
  – Complex key design
  – Ability to query data based only on combination of partition and row key
- **In-memory cache** (Redis Cache)
  + Extremely scalable, high throughput, very low response time
  – Only Key/value store, very expensive

Storage Comparison in Microsoft Azure

Database

	DATABASE THROUGHPUT UNITS	DATABASE SIZE	POINT IN TIME RESTORE	PRICE
B	5	2 GB	7 Days	$0.0067/hr (~$5/mo)

	DATABASE THROUGHPUT UNITS	DATABASE SIZE	POINT IN TIME RESTORE	PRICE
S0	10	250 GB	14 Days	$0.0202/hr (~$15/mo)
S1	20	250 GB	14 Days	$0.0403/hr (~$30/mo)
S2	50	250 GB	14 Days	$0.1008/hr (~$75/mo)
S3	100	250 GB	14 Days	$0.2016/hr (~$150/mo)

	DATABASE THROUGHPUT UNITS	DATABASE SIZE	POINT IN TIME RESTORE	PRICE
P1	100	500 GB	35 Days	$0.625/hr (~$465/mo)
P2	200	500 GB	35 Days	$1.25/hr (~$930/mo)
P3	1000	500 GB	35 Days	$5/hr (~$3,720/mo)

Redis

CACHE NAME	CACHE SIZE	BASIC	STANDARD
C0	250 MB	$0.022/hr (~$16/mo)	$0.055/hr (~$41/mo)
C1	1 GB	$0.055/hr (~$41/mo)	$0.138/hr (~$103/mo)
C2	2.5 GB	$0.09/hr (~$67/mo)	$0.225/hr (~$167/mo)
C3	6 GB	$0.18/hr (~$134/mo)	$0.45/hr (~$335/mo)
C4	13 GB	$0.21/hr (~$156/mo)	$0.525/hr (~$391/mo)
C5	26 GB	$0.42/hr (~$312/mo)	$1.05/hr (~$781/mo)
C6	53 GB	$0.84/hr (~$625/mo)	$2.10/hr (~$1,562/mo)

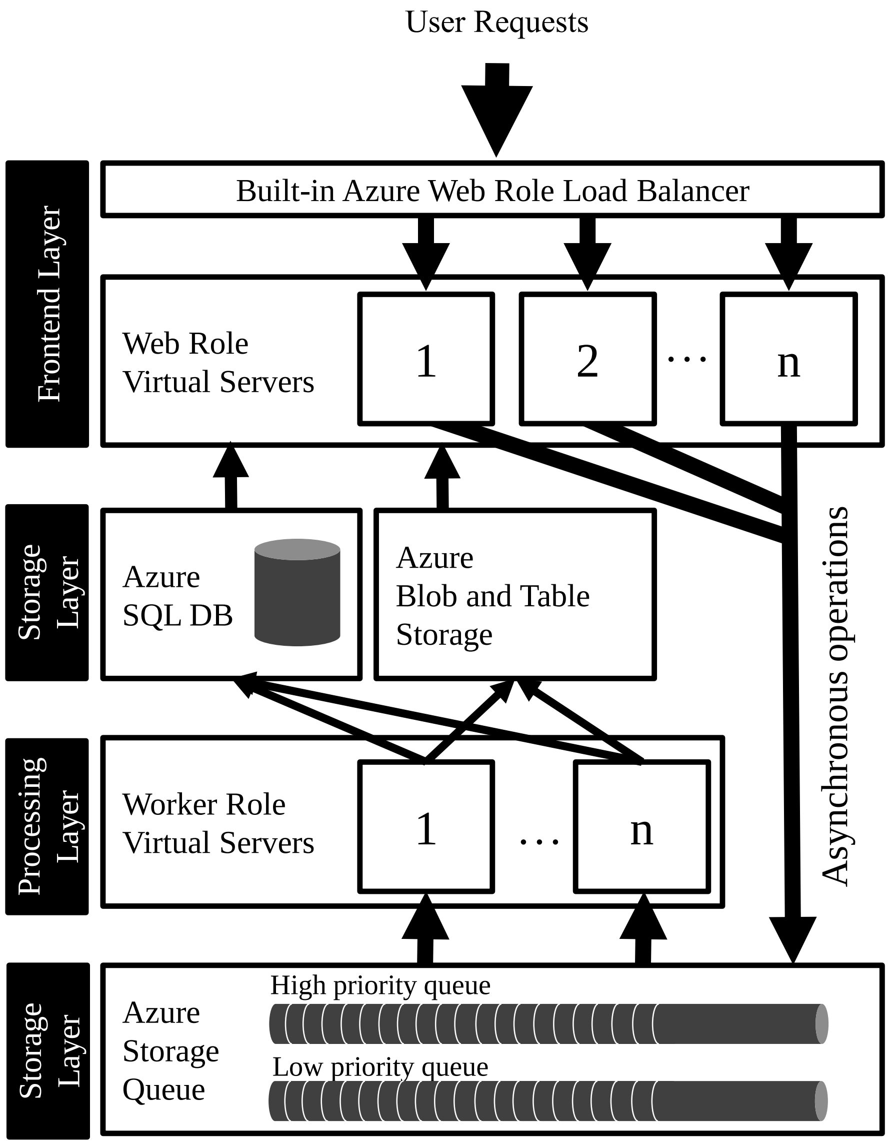
Azure Storage

STORAGE CAPACITY	LRS	GRS
First 1 TB / Month	$0.07 per GB	$0.095 per GB
Next 49 TB (1 to 50 TB) / Month	$0.065 per GB	$0.08 per GB
Next 450 TB (50 to 500 TB) / Month	$0.06 per GB	$0.07 per GB

+$0.0036 per 100,000 transactions

2. Indirect Dependency Tactic

- User requests are not processed synchronously be the web server
- Task requests are stored in a scalable queue service
- Number of running workers is variable (cost effective)
- Results are stored in a scalable storage service
- **Problem:** How to notify user about changes or errors in processing?



3. Results Pre-computation Tactic

- Effective combination of previously mentioned tactics:
  - **Multi-tiered Storage Tactic** – provides cost effective scalable storage
  - **Indirect Dependency Tactic** – provides spare compute resources

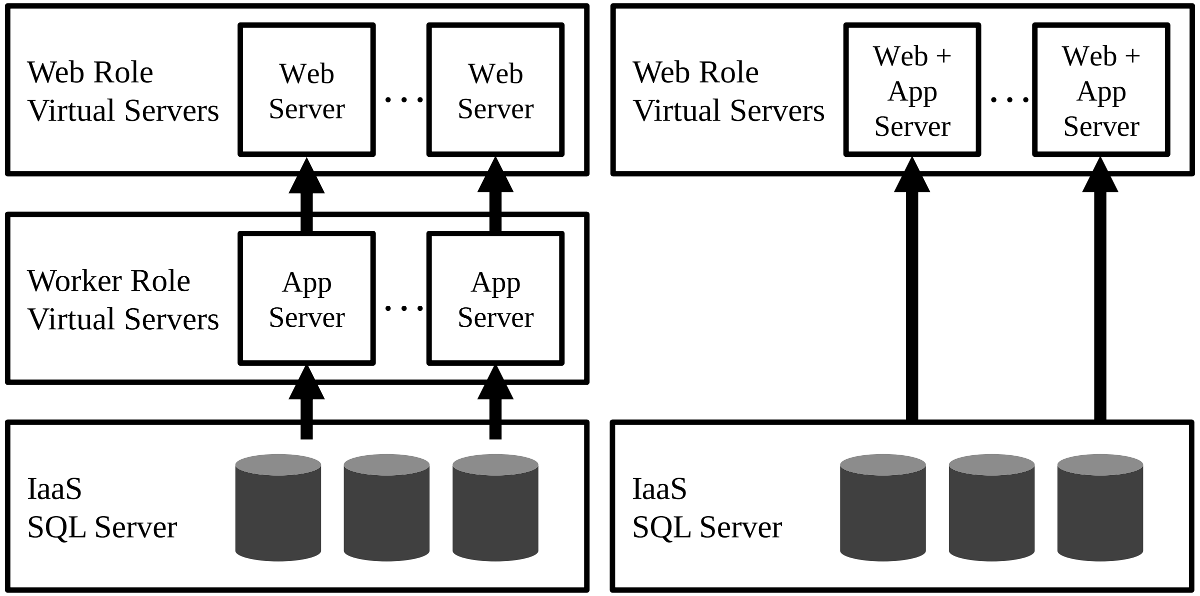
- **Goal:** Increase cost effectivity of reserved compute resources

- Workers are billed based on their uptime independently on their CPU load
- Results are stored in a form which does not require additional processing

4. Component Co-deployment Tactic

- **Goal:** Minimize inter role communication latency

- In the cloud communication latency between different services may be significantly higher than in on-premise environment
- Necessary to minimize service calls
- **Apply Affinity Groups**



Outline

1. Introduction to a cloud environment and its foundations
2. Identification of relevant software quality attributes in the cloud
3. Frequent mistakes in cloud application design
4. Cloud specific architectural tactics and guidelines for their application