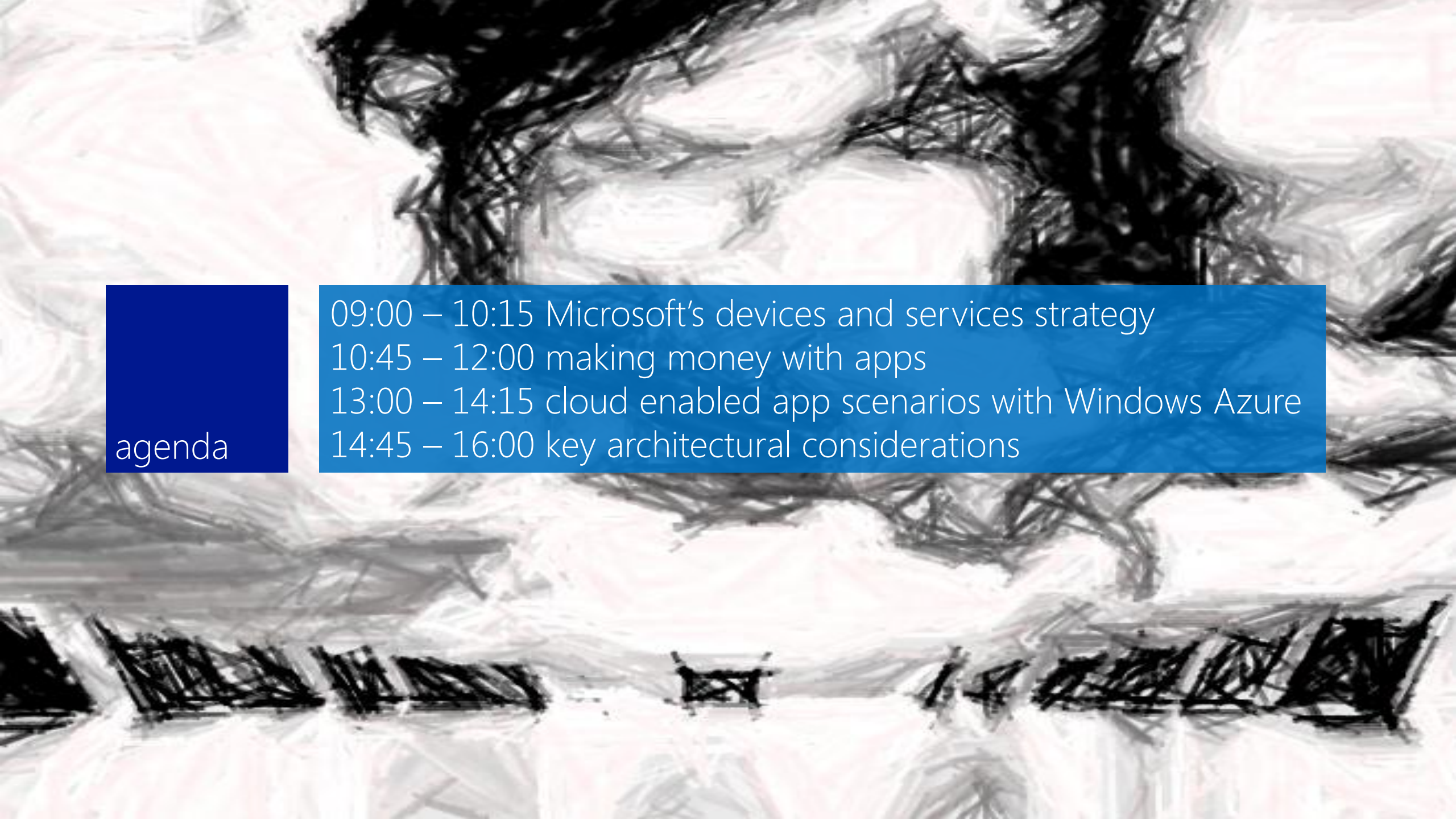
cloud strategy day

"How the Microsoft cloud enables the best App experiences across devices"

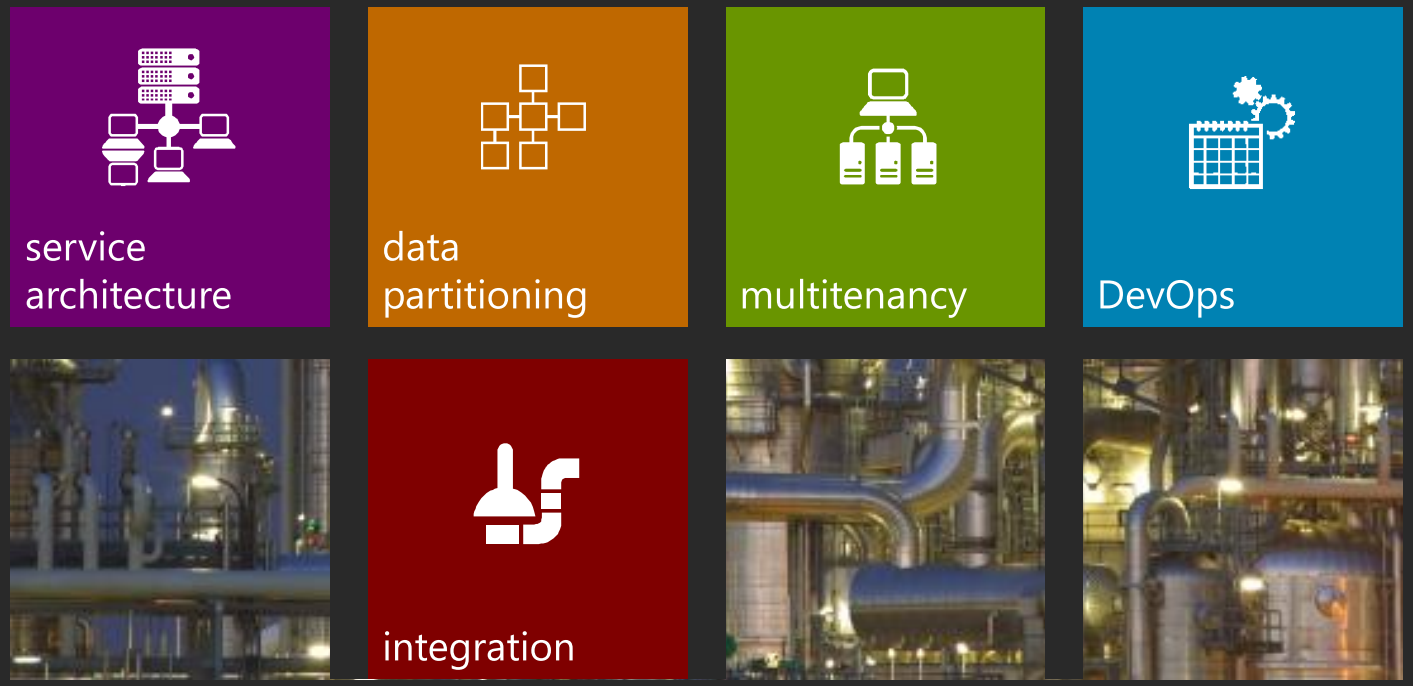
Beat Schwegler – beatsch@microsoft.com
Director, Platform Strategy Group, Microsoft Corp.



agenda

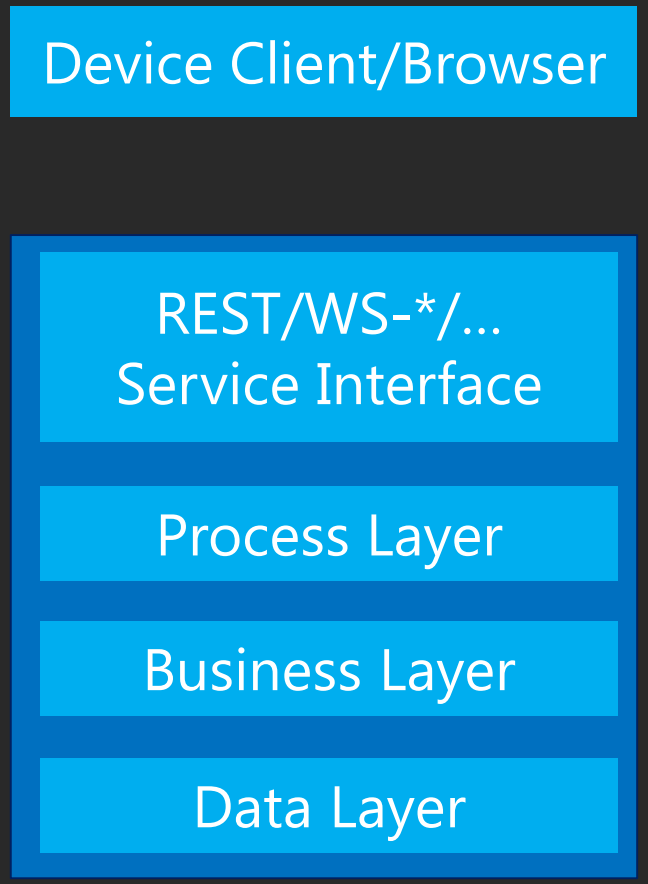
09:00 – 10:15 Microsoft's devices and services strategy
10:45 – 12:00 making money with apps
13:00 – 14:15 cloud enabled app scenarios with Windows Azure
14:45 – 16:00 key architectural considerations

chapter IV

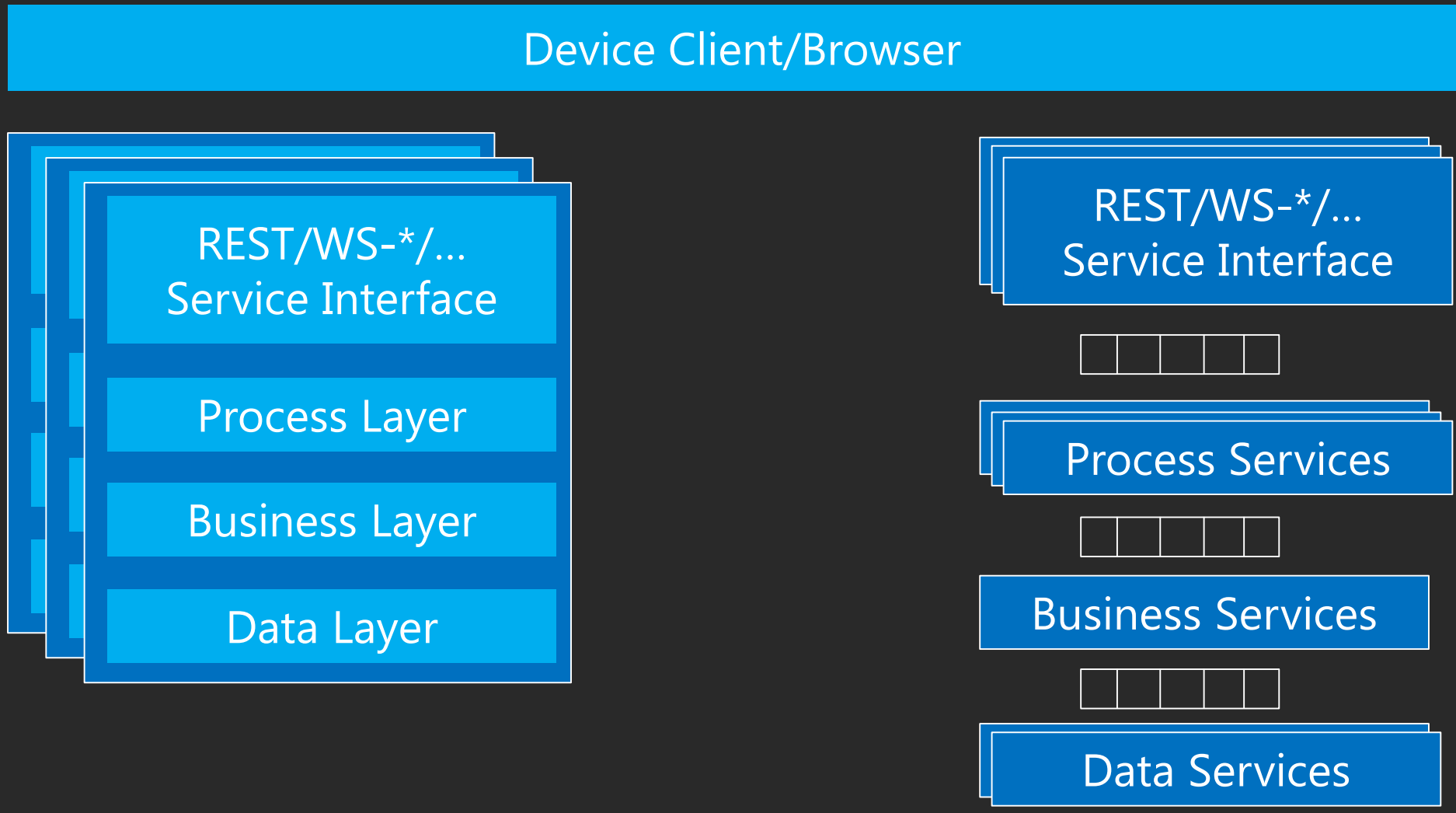

service architecture
data partitioning
multitenancy
DevOps
integration


service
architecture

service architecture

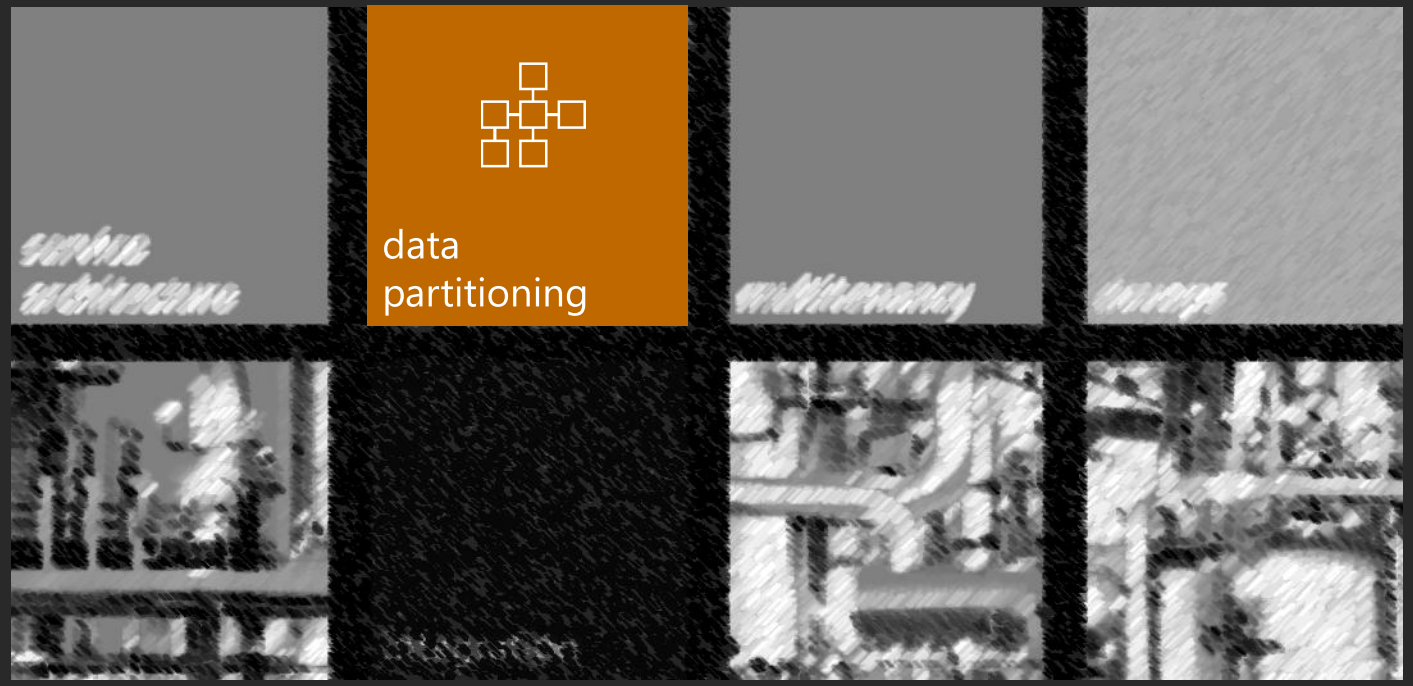


service architecture

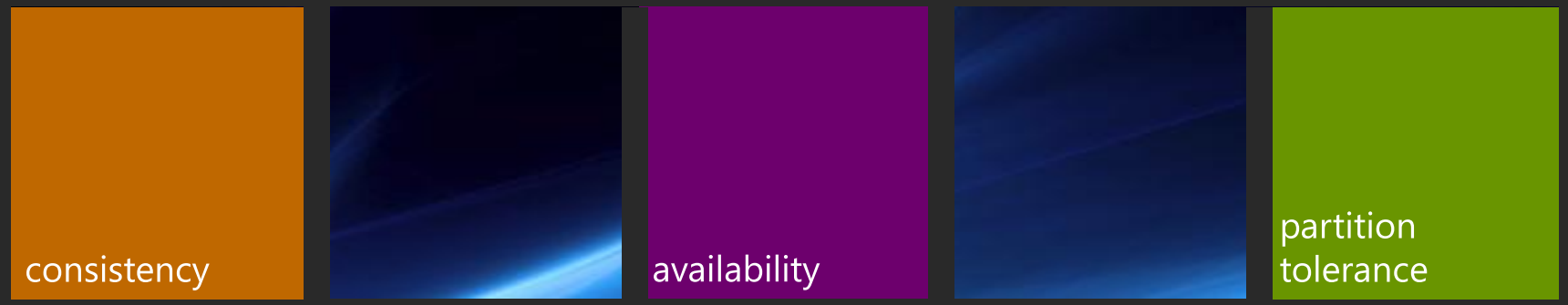


each instance contains all layers
inproc communication

instance per layer, communication through queues


data
partitioning

cap theorem



data partitioning

traditional reasons

data volume (too many bytes)

work load (too many transactions/second)

new 'cloud focused' reasons

cost (using different cost storage)

elasticity (just in time partitioning for high load periods)

horizontal partitioning

First Name	Last Name	Email	Thumbnail	Photo
David	Alexander	davida@contoso.com	3kb	3MB
Jared	Carlson	jaredc@contoso.com	3kb	3MB
Sue	Charles	suec@contoso.com	3kb	3MB
Simon	Mitchel	simonm@contoso.com	3kb	3MB
Richard	Zeng	richardz@contoso.com	3kb	3MB

horizontal partitioning (sharding)

spread data across similar nodes

achieve massive scale out (data and load)

intra-partition queries are simple

cross-partition queries are harder

vertical partitioning

First Name	Last Name	Email	Thumbnail	Photo
David	Alexander	davida@contoso.com	3kb	3MB
Jared	Carlson	jaredc@contoso.com	3kb	3MB
Sue	Charles	suec@contoso.com	3kb	3MB
Simon	Mitchel	simonm@contoso.com	3kb	3MB
Richard	Zeng	richardz@contoso.com	3kb	3MB

SQL Azure

Tables

BLOBS

vertical partitioning

place frequently queried data in more ‘expensive’ indexed storage

retrieving a whole row requires >1 query

spread data across dis-similar nodes

place large data in ‘cheap’ binary storage

hybrid partitioning

First Name	Last Name	Email	Thumbnail	Photo
David	Alexander	davida@contoso.com	3kb	3MB
Jared	Carlson	jaredc@contoso.com	3kb	3MB
Sue	Charles	suec@contoso.com	3kb	3MB
Simon	Mitchel	simonm@contoso.com	3kb	3MB
Richard	Zeng	richardz@contoso.com	3kb	3MB

tables != rdbms

cross partition queries are resource intensive

aggressive data duplication can save money and boost performance

storage is cheap

goal: To be able to include Partition Key in all queries

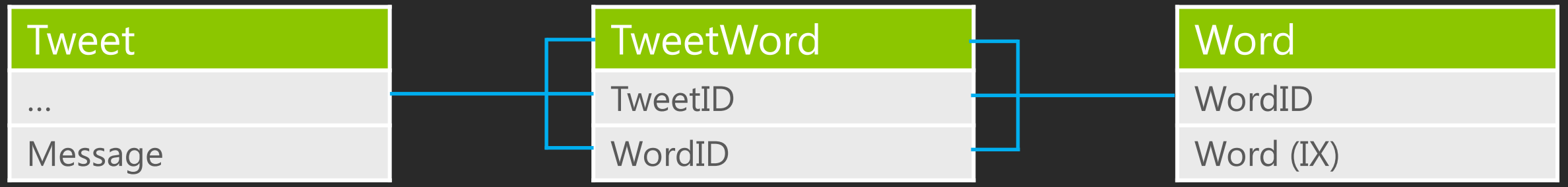
e.g. tweet storage

Tweet
TweetID
UserID
DateTimeStamp
Message

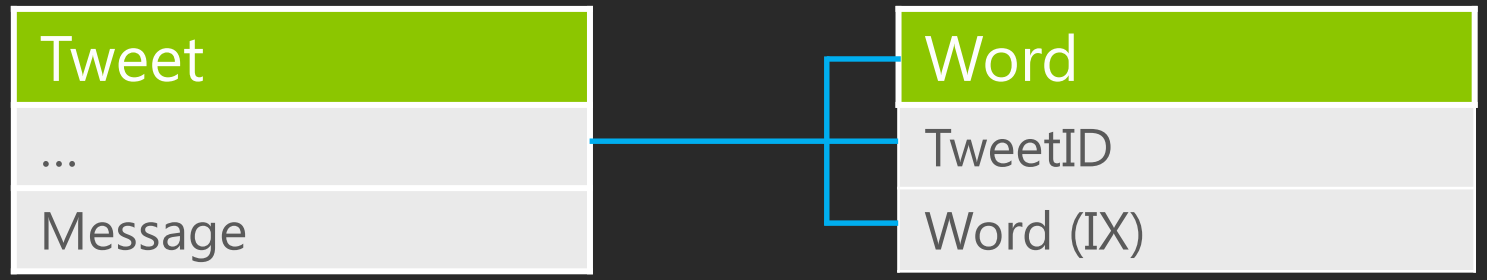
With an RDBMS you'd probably start something like this:
SELECT * FROM Tweet WHERE Message Like %SearchTerm%

e.g. tweet storage

You'd soon realize that LIKE isn't so wonderful.
You'd do a little normalization



Which quickly becomes this as len(key) approaches AVG len(word)



e.g. tweet storage

With Tables we go the whole way

Tweet
TweetID (RK)
UserID (PK)
DateTimeStamp
Message

Worker Role Creates

TweetIndex
TweetID (RK)
UserID
DateTimeStamp
Message
Word (PK)

GET All Entities in Partition ‘DavidA’ from Tweet

GET All Entities in Partition ‘Foo’ from TweetIndex

e.g. tweet storage

We may create multiple indexes

Tweet
TweetID (RK)
UserID (PK)
DateTimeStamp
Message

Worker Role Creates

MentionIndex
TweetID (RK)
UserID
DateTimeStamp
Message
MentionedUserID (PK)

GET All Entities in Partition ‘DavidA’ from MentionIndex

traffic management fundamentals

performance

directs the user to the "best"/"closest" deployment

example:
direct the user to the "best" deployment between US South and West Europe

failover

one deployment is primary

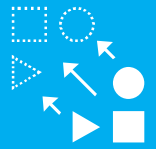
traffic is redirected to another deployment if the primary goes down

example:
all traffic is directed to US North; if it goes down, send all traffic to US South

geomapping

allows users from defined geographic locations to be directed to particular deployment

example:
all users from US -> US North, all users from Asia -> US North, all users from Europe -> West Europe



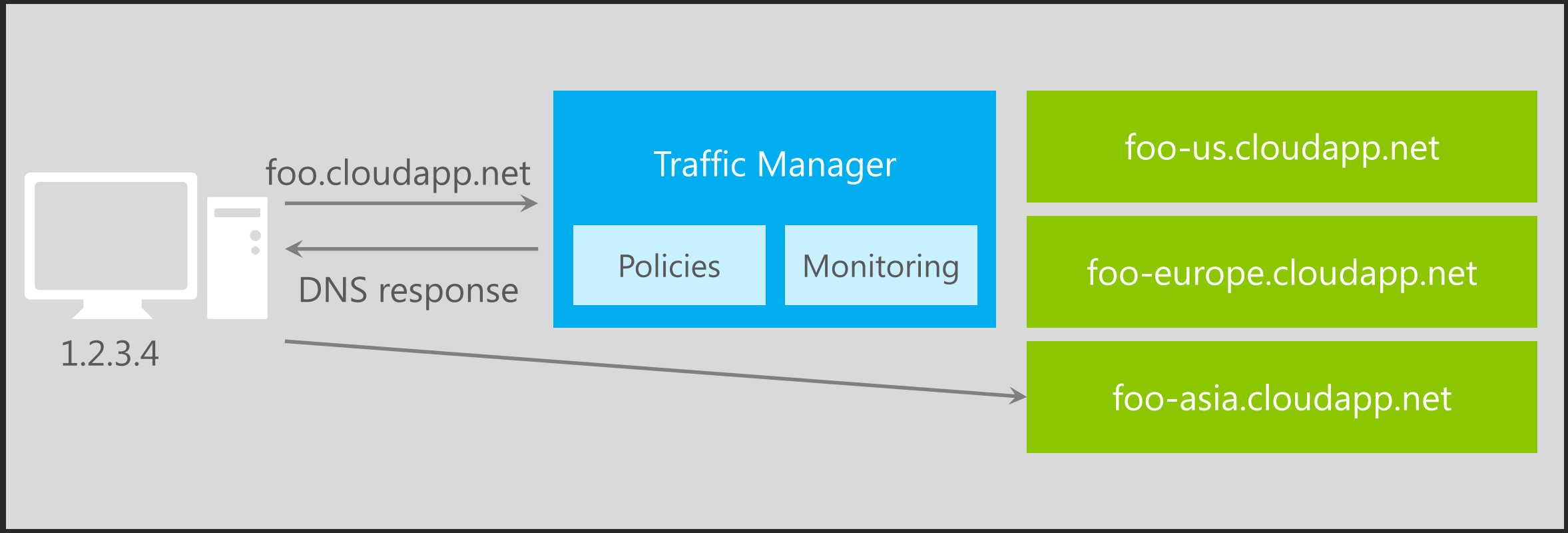
ratio

sends traffic to different deployments based on fixed ratio (N/M)

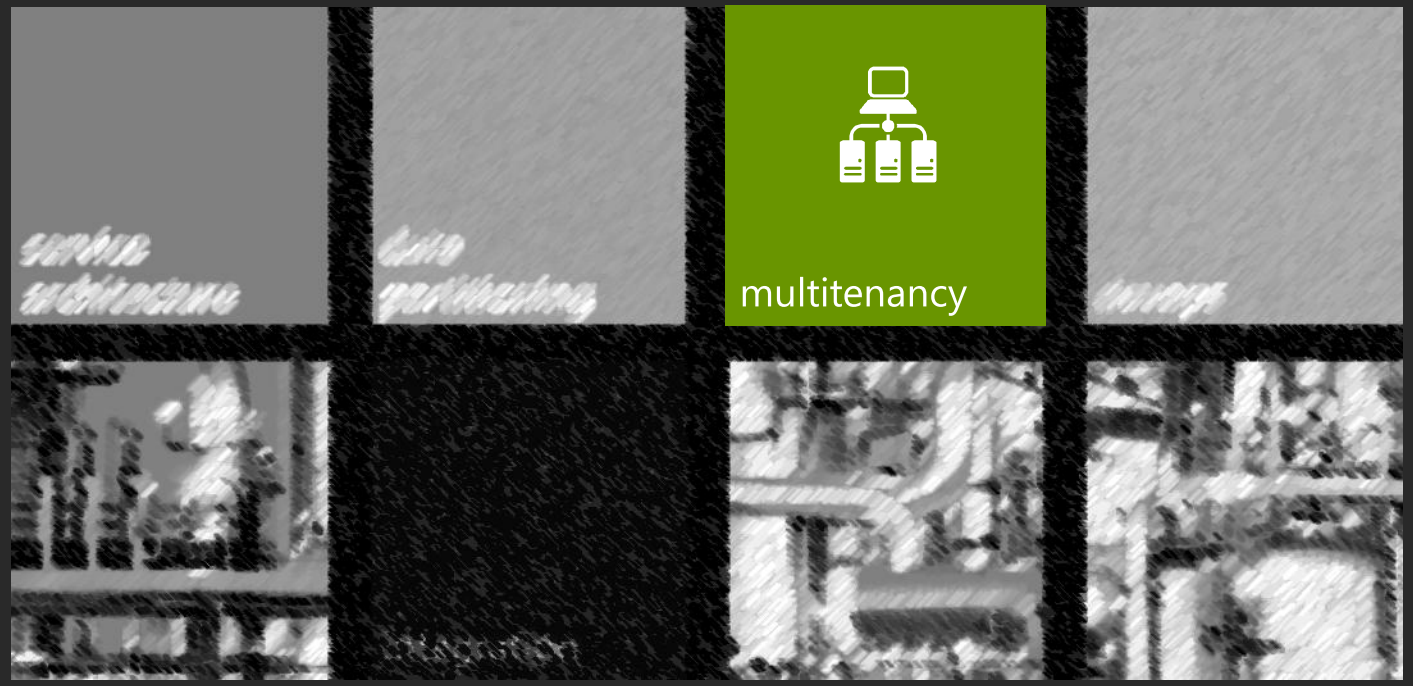
example:
direct 20% of user traffic to US South and 80% to US North.

Windows Azure Traffic Manager

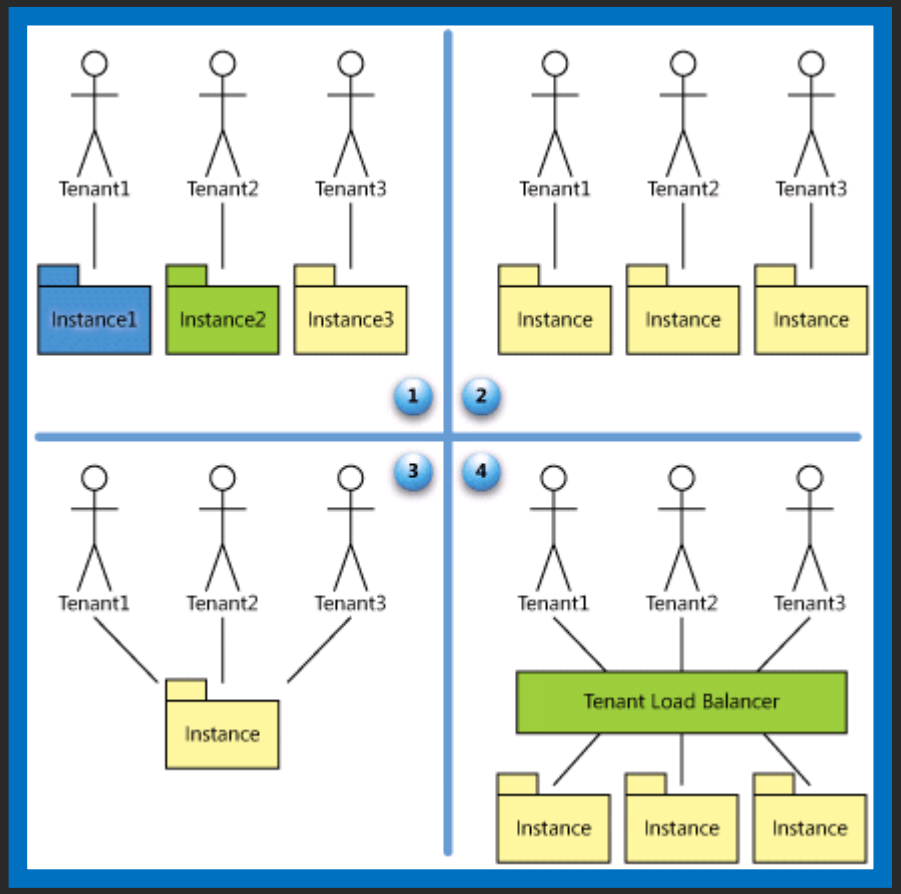
Direct users to the service in the closest region
with the Windows Azure Traffic Manager




demo:
Traffic Manager


multitenancy

multitenancy




DevOps

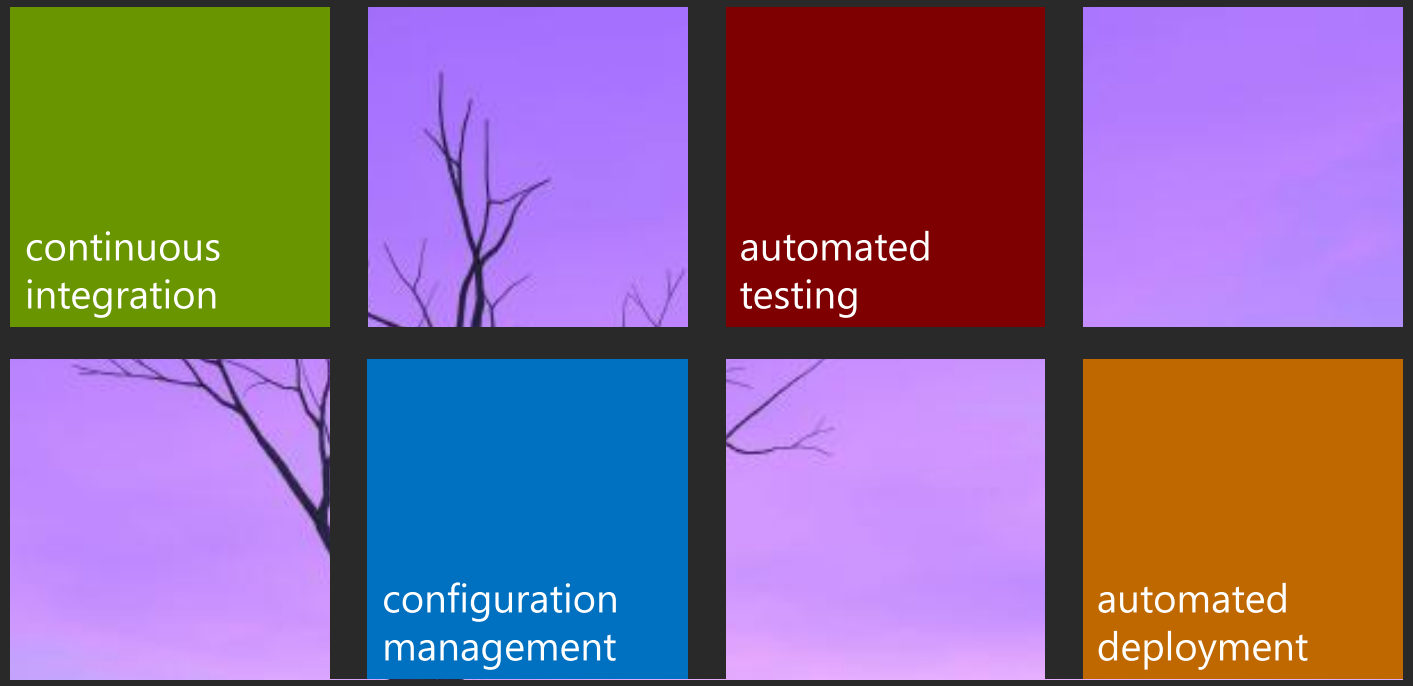
continues deployment

http://code.flickr.com/

devops

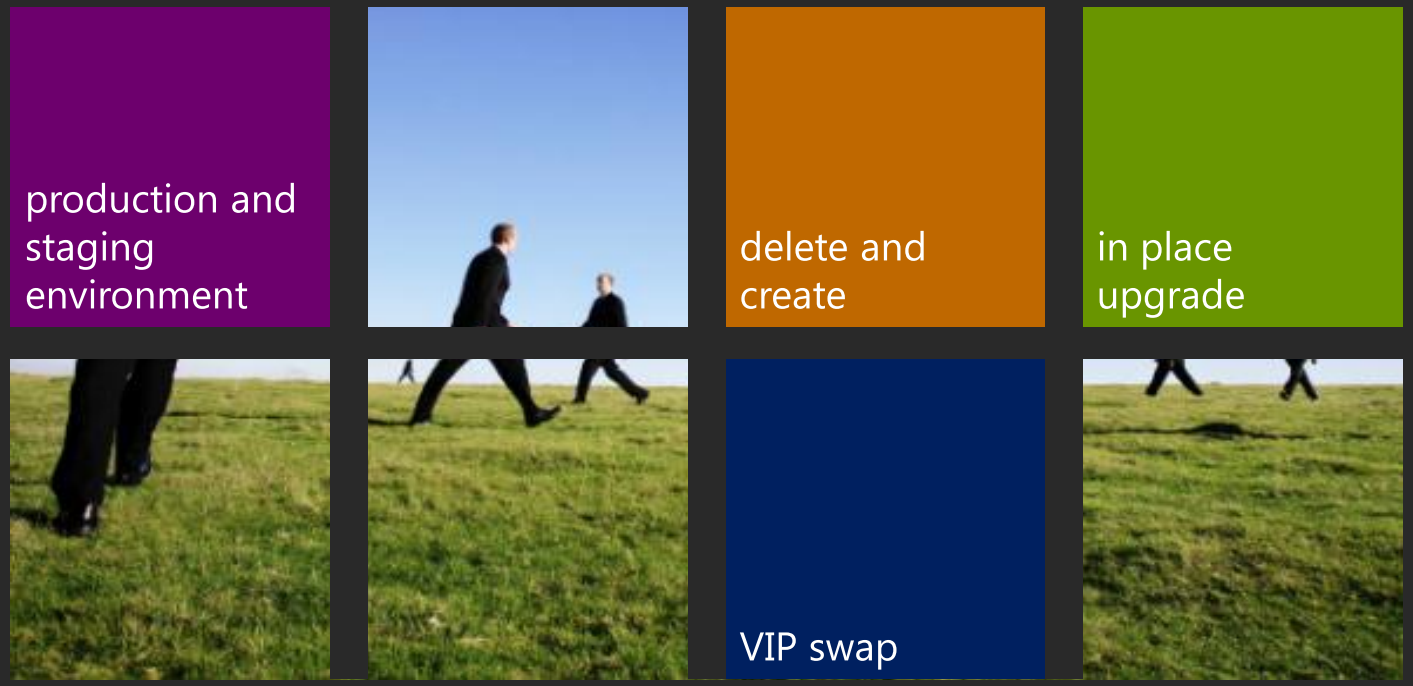

development
quality assurance
deployment
operations
one team
continuous delivery

devops

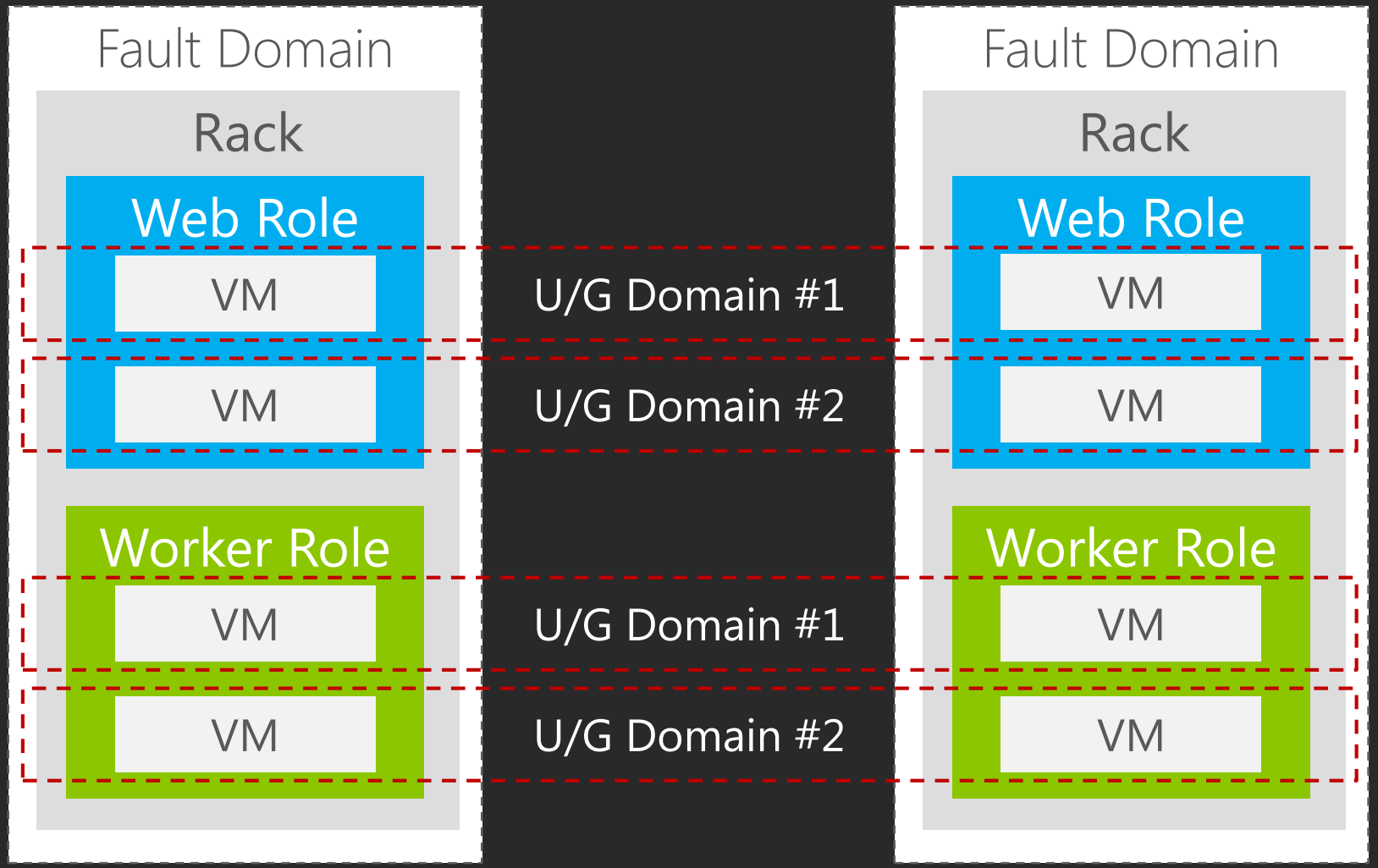

continuous integration
automated testing
configuration management
automated deployment

deployment/release strategies

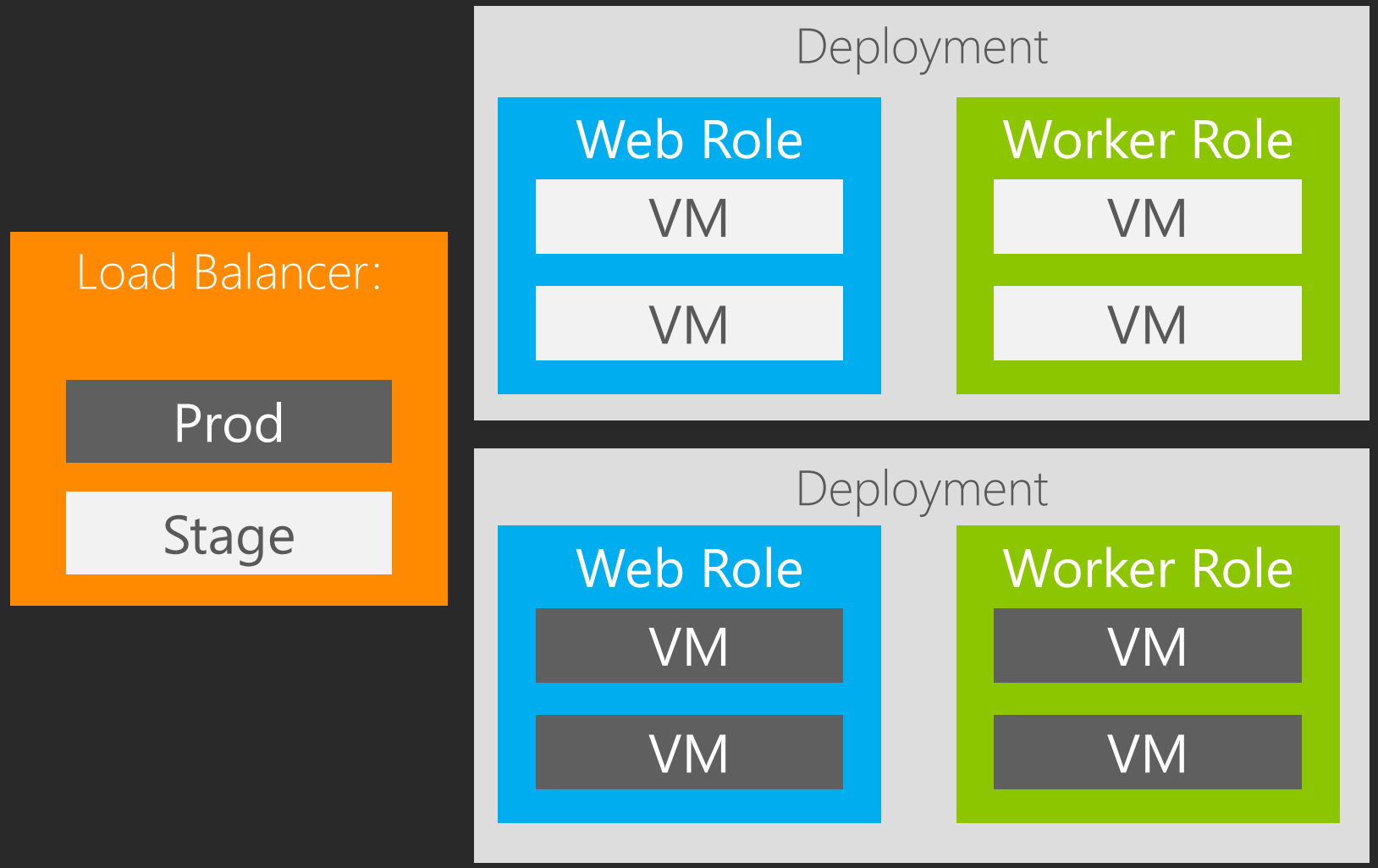
upgrades in Windows Azure



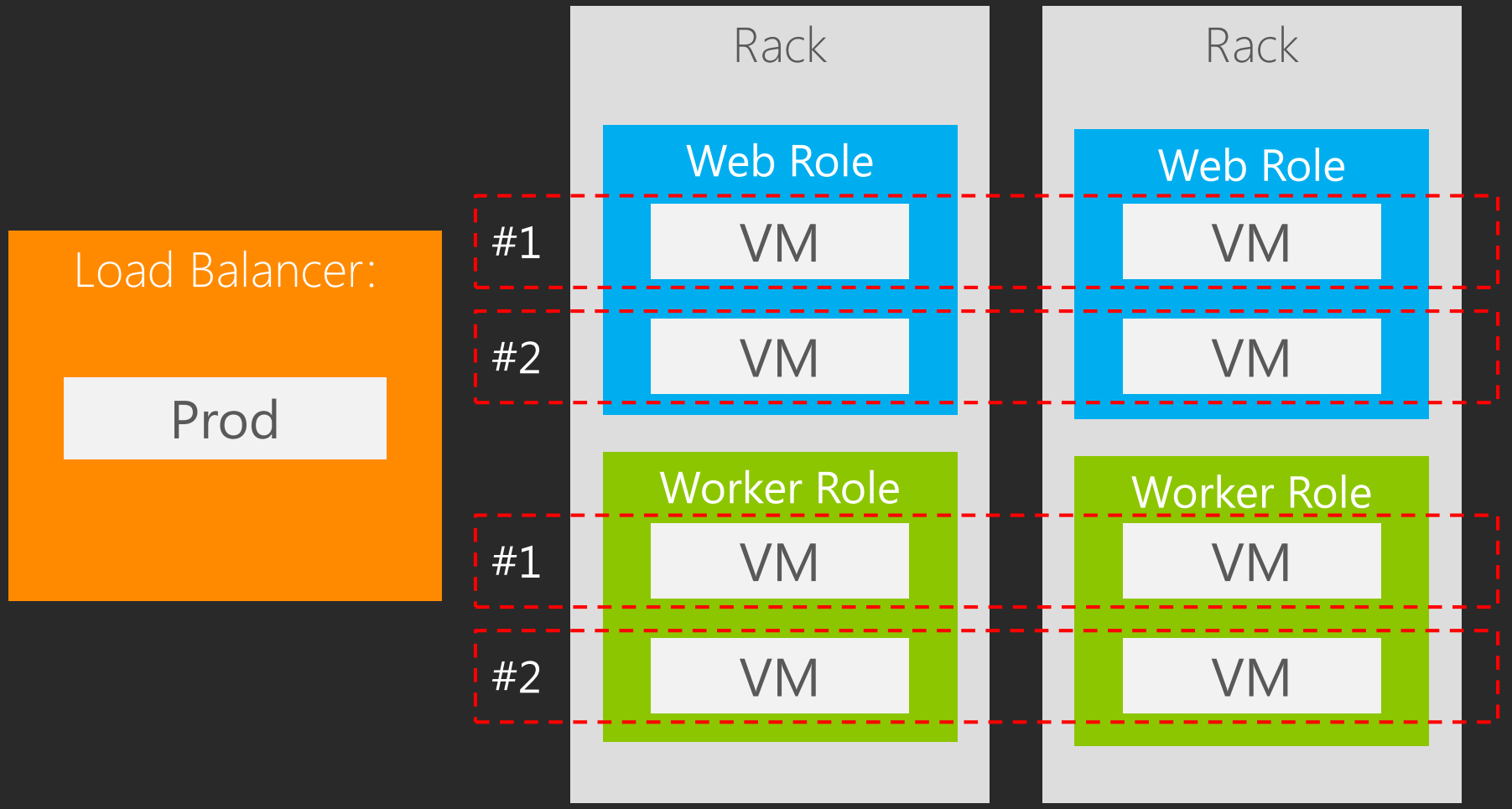
fault and upgrade domains



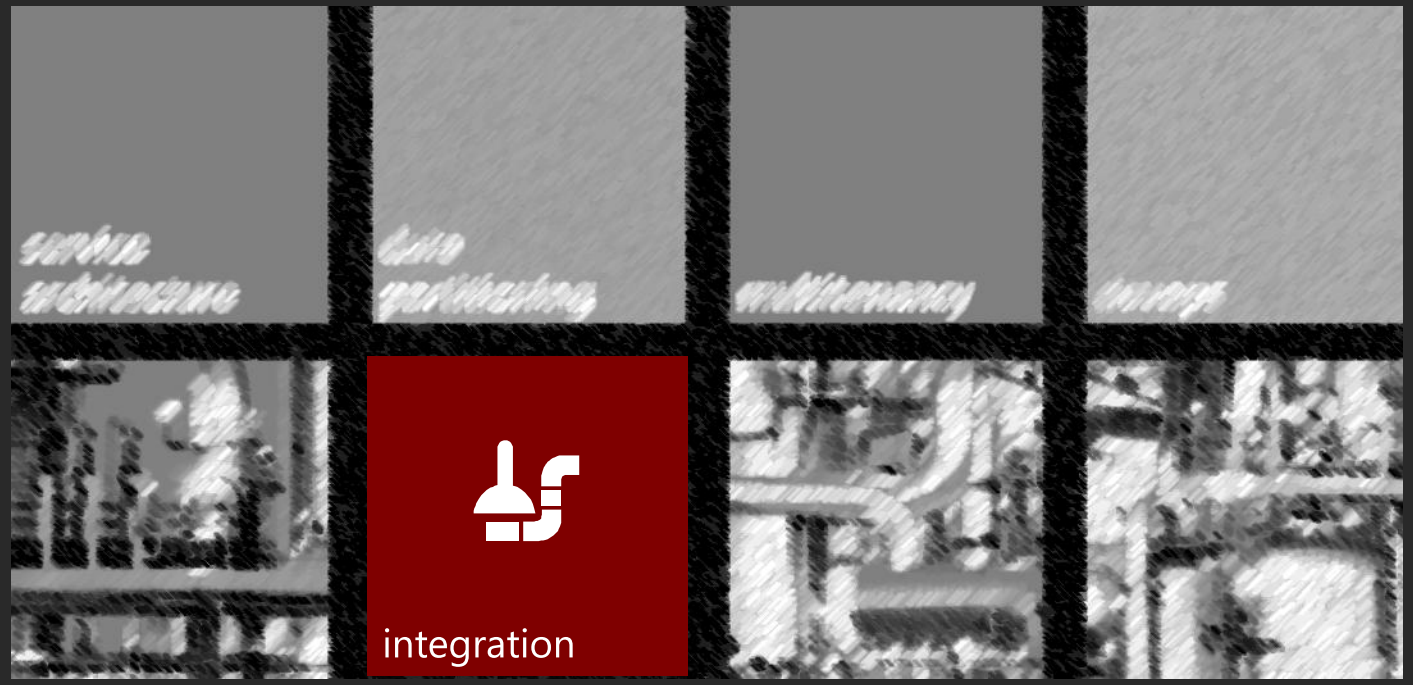
VIP swap


Load Balancer:
Prod
Stage
Deployment
Web Role
VM
VM
Worker Role
VM
VM
Deployment
Web Role
VM
VM
Worker Role
VM
VM

in place upgrade



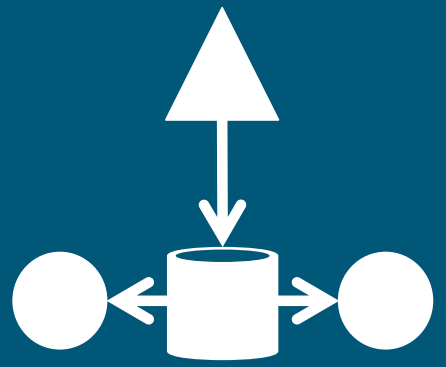
Windows Azure deployment tools


integration

Windows Azure Service Bus

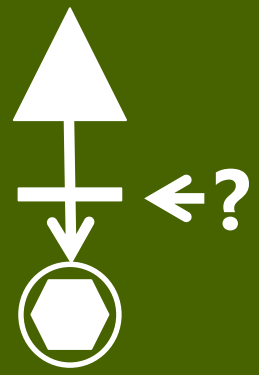
Connectivity
Service Relay
Protocol Tunnel
Eventing, Push

Rich options for interconnecting apps across network boundaries



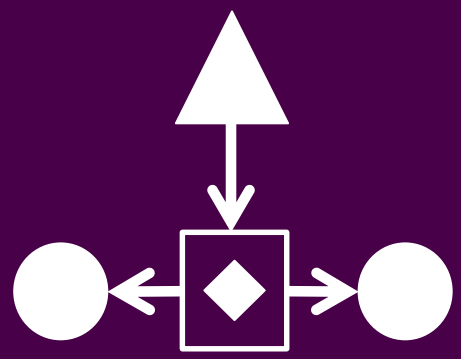
Messaging
Queuing
Pub/Sub
Reliable Transfer

Reliable, transaction-aware cloud messaging infrastructure for business apps



Service
Management Naming,
Discovery
Monitoring

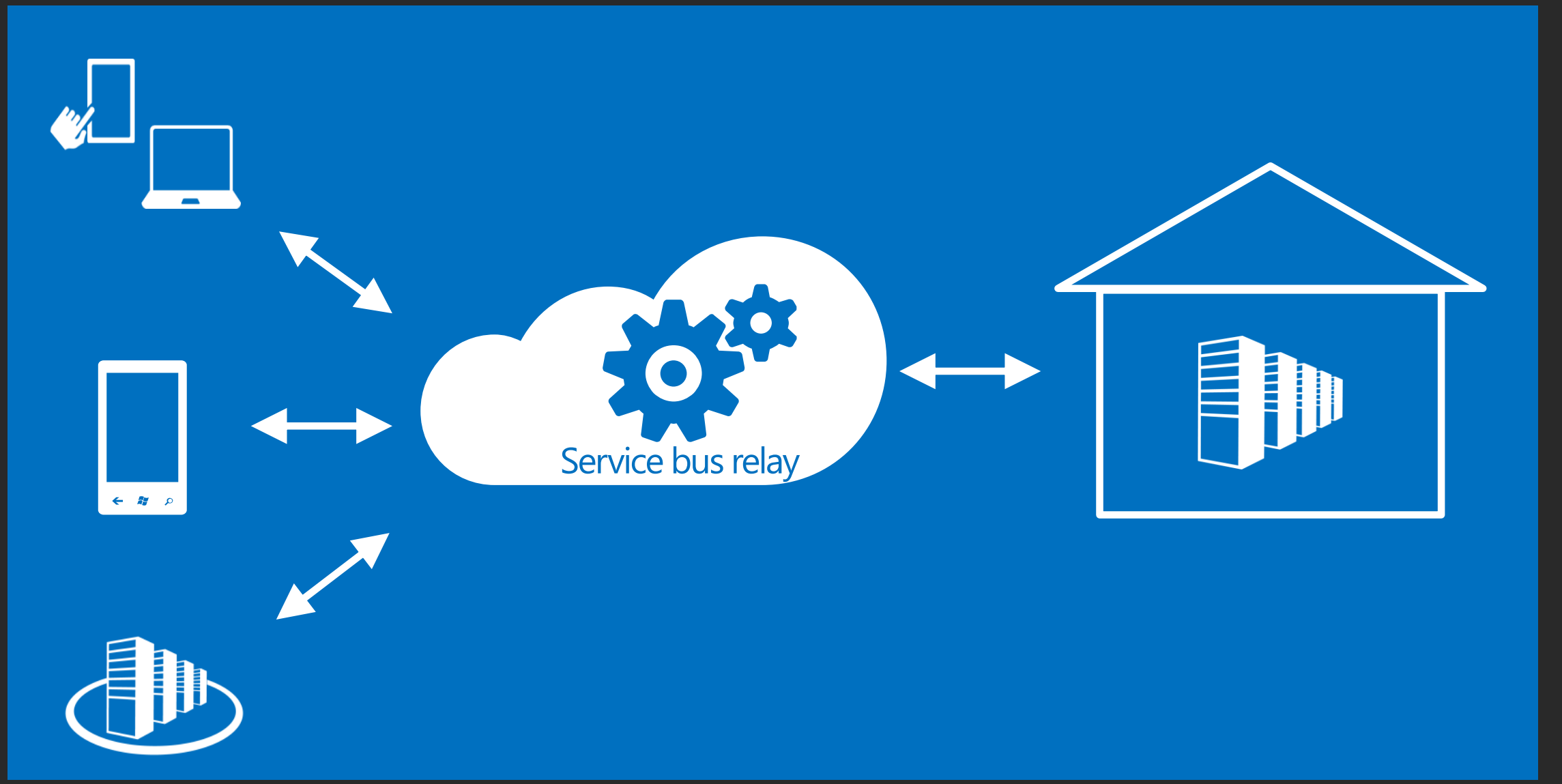
Consistent management surface and service observation capabilities



Integration
Routing
Coordination
Transformation

Content-based routing, document transformation, and process coordination

accessing on-premise services




Demo:
Service Bus
Relay

conclusion

cloud style computing designs for scale out

scale out requires partitioning

having redundant/duplicated data is ok

continuous delivery is a key asset

chapter V

closing

identity

communication

lifecycle

1 solution
many Apps
1 scenario

1 solution
1 application
many scenarios

storage

monetization