



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.


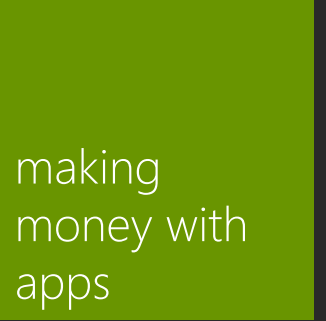
agenda



Microsoft's
devices and
services
strategy



making
money with
apps



cloud enabled
app scenarios
with Windows
Azure



key
architectural
considerations

chapter IV



service
architecture



data
partitioning



multitenancy



DevOps



integration





service
architecture

data
partitioning

multitenancy

storage

integration

service architecture

Device Client/Browser

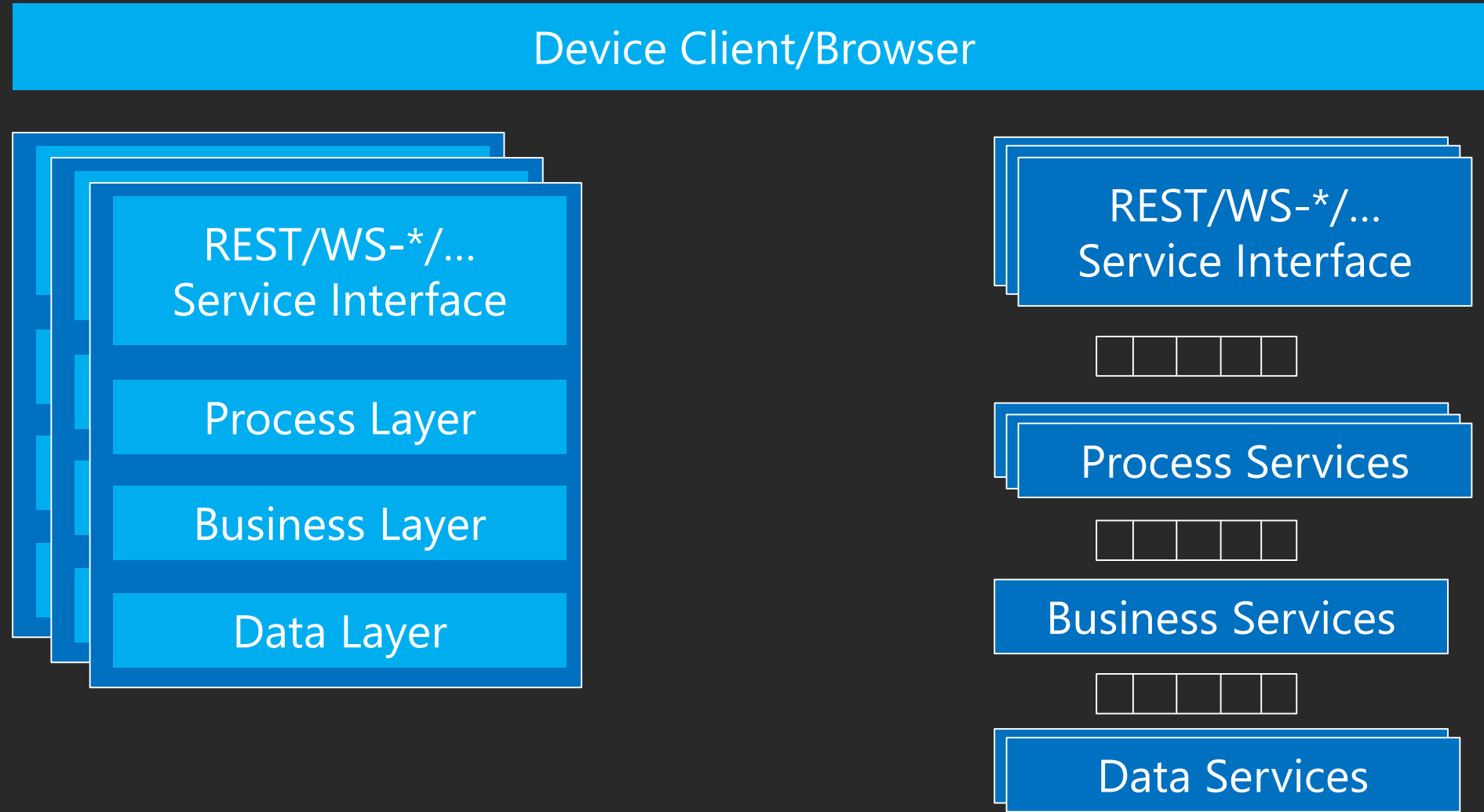
REST/WS-*/...
Service Interface

Process Layer

Business Layer

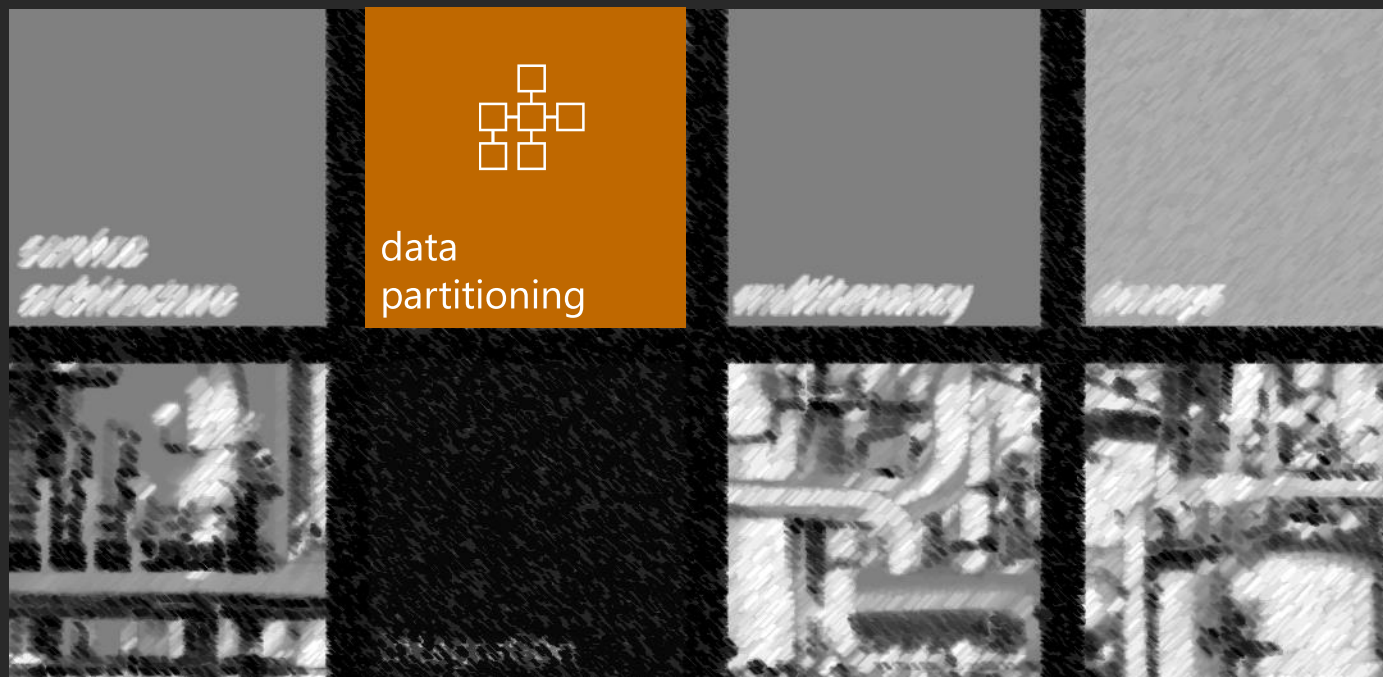
Data Layer

service architecture

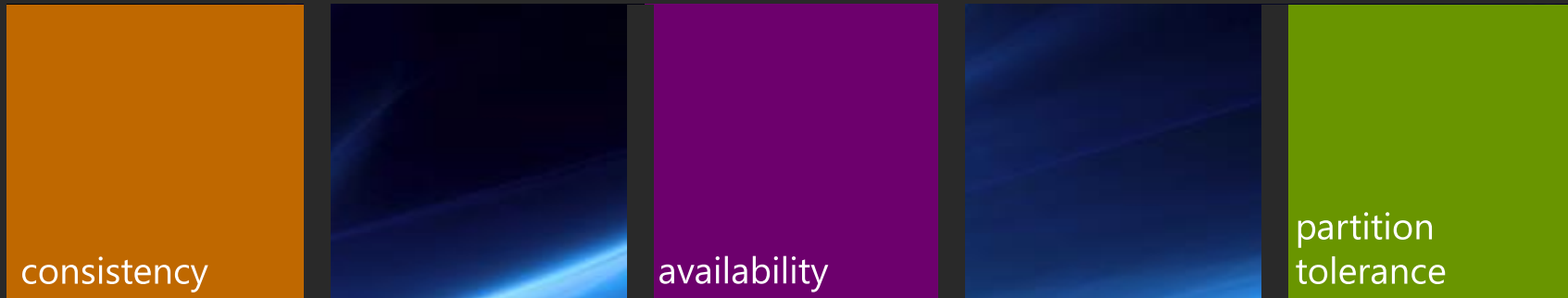


each instance contains all layers
inproc communication

instance per layer, communication through queues




cap theorem



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



The diagram illustrates horizontal partitioning of a table. The table is divided into four horizontal sections, each representing a partition. Each partition is color-coded and contains one or more rows. Arrows point from the rows to server icons below, indicating that each partition is stored on a separate server.

- Blue Partition:** Contains the row for David Alexander (davida@contoso.com). An arrow points from this row to a server icon.
- Orange Partition:** Contains the rows for Jared Carlson (jaredc@contoso.com) and Sue Charles (suec@contoso.com). An arrow points from the first row of this partition to a server icon.
- Purple Partition:** Contains the row for Simon Mitchel (simonm@contoso.com). An arrow points from this row to a server icon.
- Red Partition:** Contains the row for Richard Zeng (richardz@contoso.com). An arrow points from this row to a server icon.

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

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

The diagram illustrates hybrid partitioning. The table is partitioned into two groups: one with a blue border (David, Jared, Sue) and one with an orange border (Simon, Richard). The Thumbnail column is highlighted with a purple border, and the Photo column is highlighted with a red border. Arrows point from the partitioned rows to server icons below. Specifically, blue arrows point from the blue-bordered rows to a server icon, and orange arrows point from the orange-bordered rows to another server icon. Purple and red arrows point from the Thumbnail and Photo columns respectively to server icons.

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

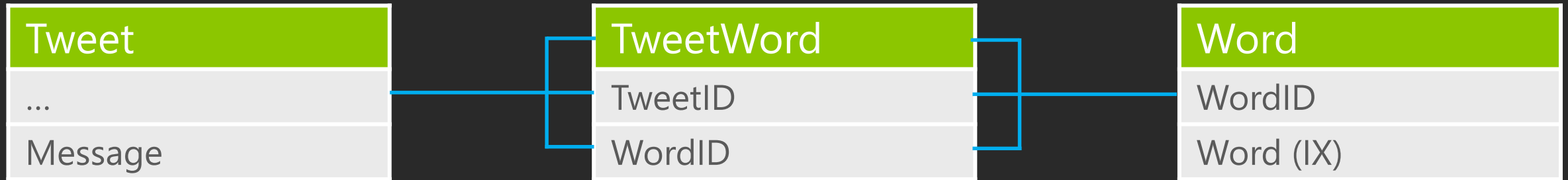
Tweet
TweetID
UserID
DateTimeStamp
Message

With an RDBMS you'd probably start something like this:

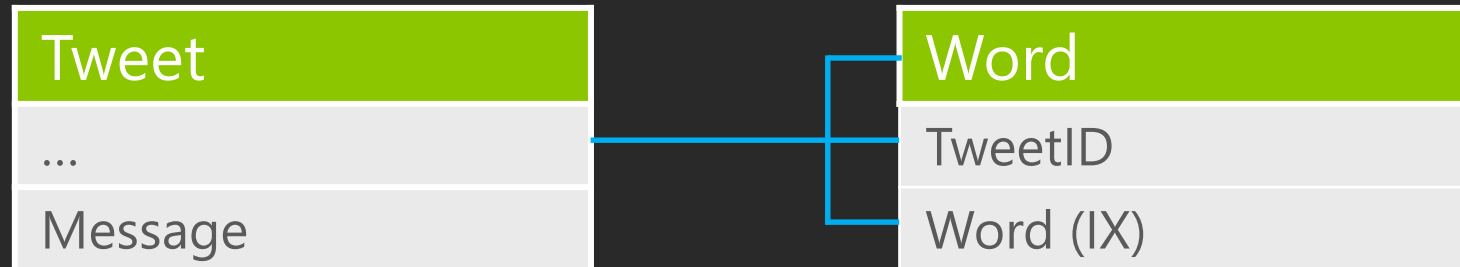
```
SELECT * FROM Tweet WHERE  
Message Like %SearchTerm%
```


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 $\text{len}(\text{key})$ approaches $\text{AVG len}(\text{word})$



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

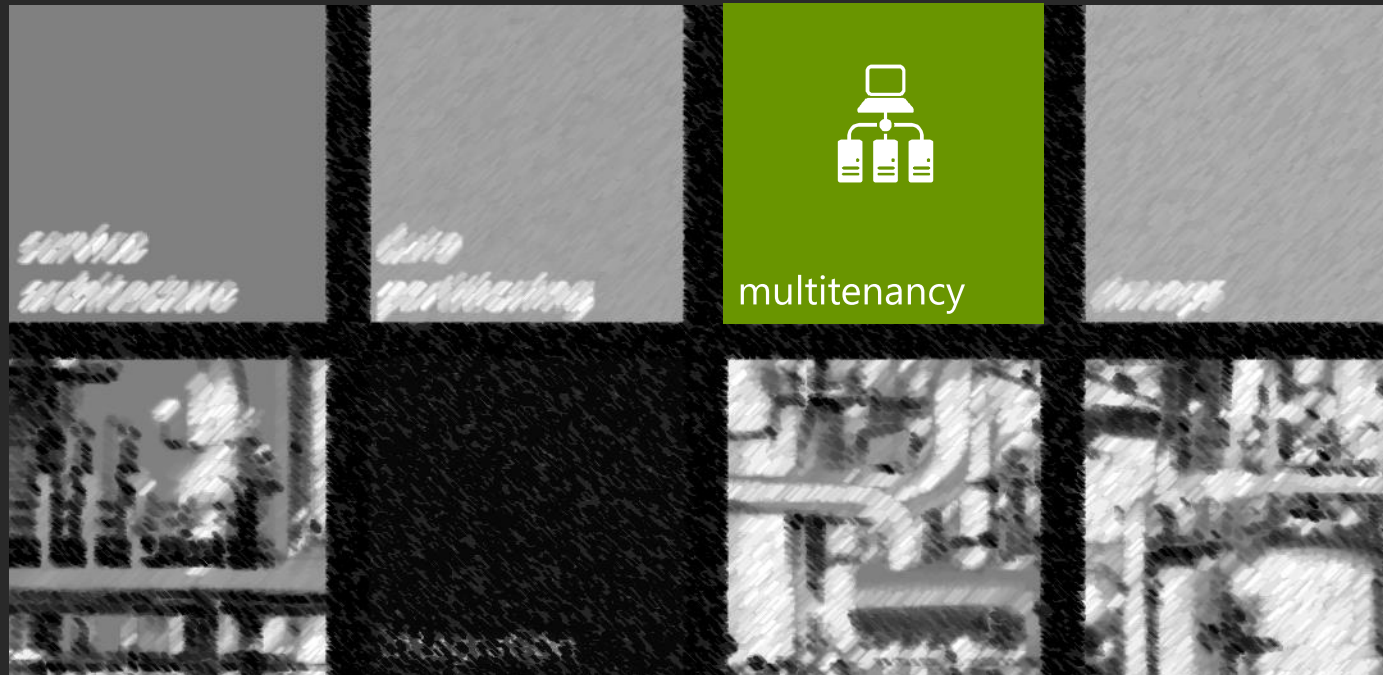
We may create multiple indexes

Tweet
TweetID (RK)
UserID (PK)
DateTimeStamp
Message

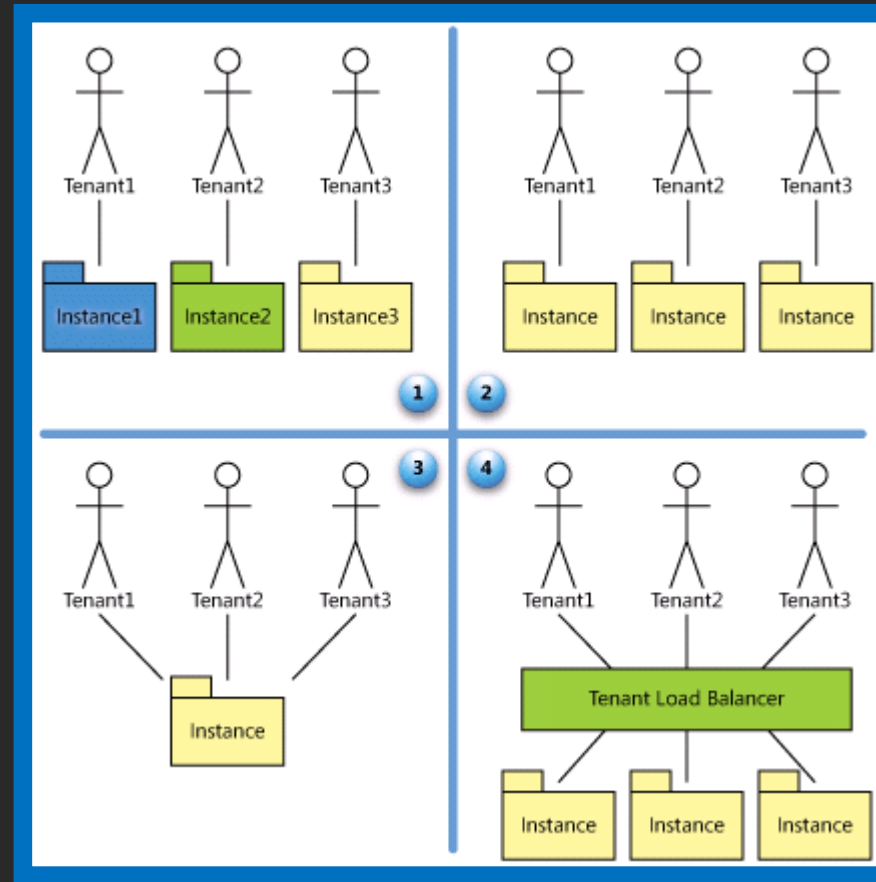
Worker Role Creates

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

GET All Entities in Partition 'DavidA' from MentionIndex



multitenancy



service
architecture

data
partitioning

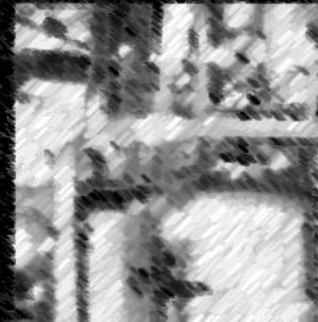
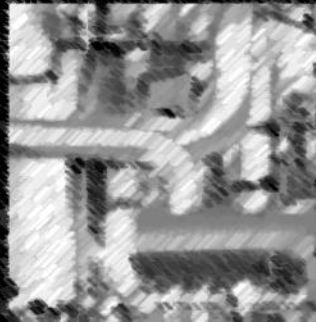
multitenancy



DevOps



cloud
migration



continues deployment

FEATURING



Flickr was last deployed 2 days ago, including 4 changes by 1 person.

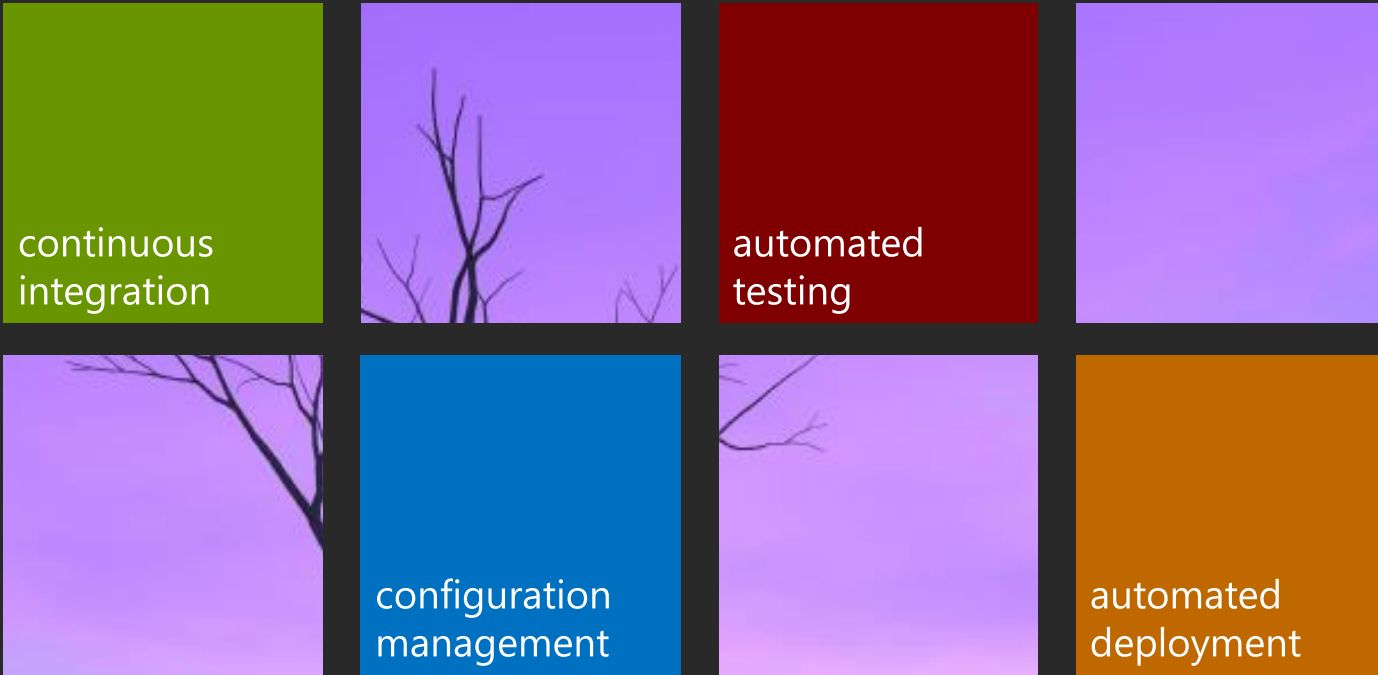
In the last week there were 77 deploys of 440 changes by 18 people.

<http://code.flickr.com/>

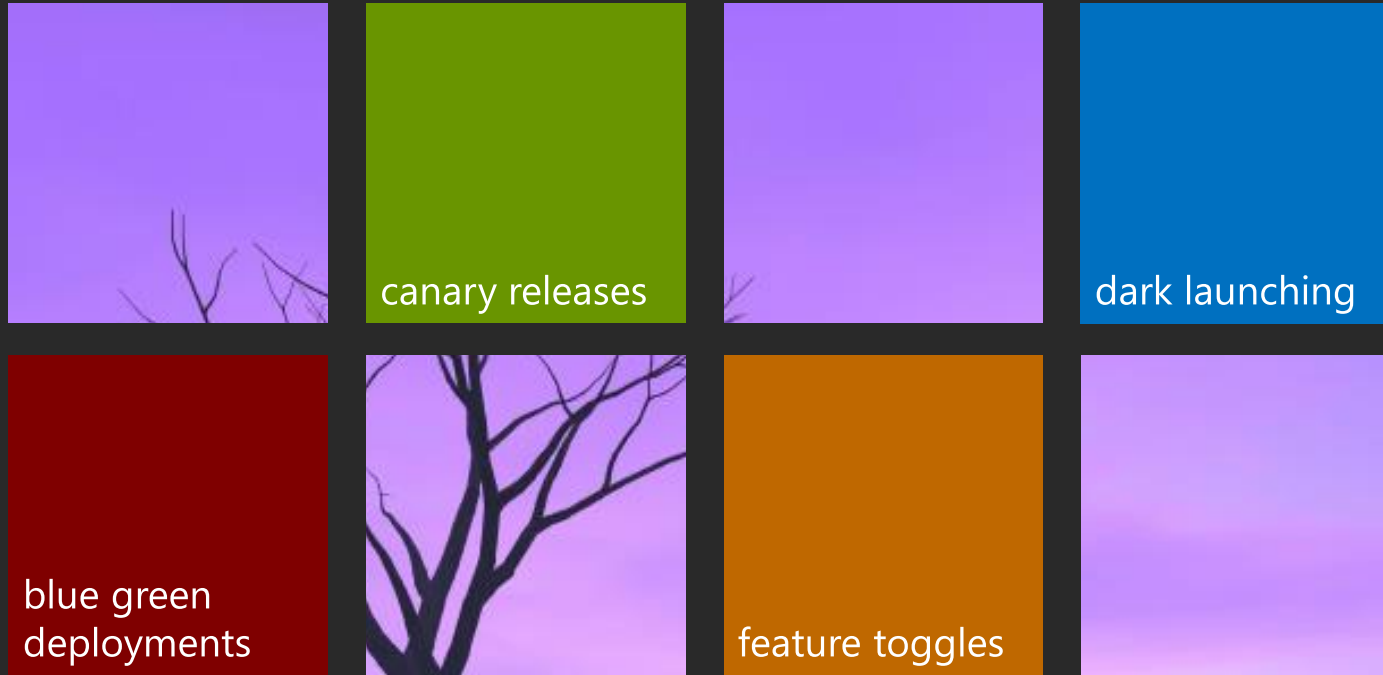
devops



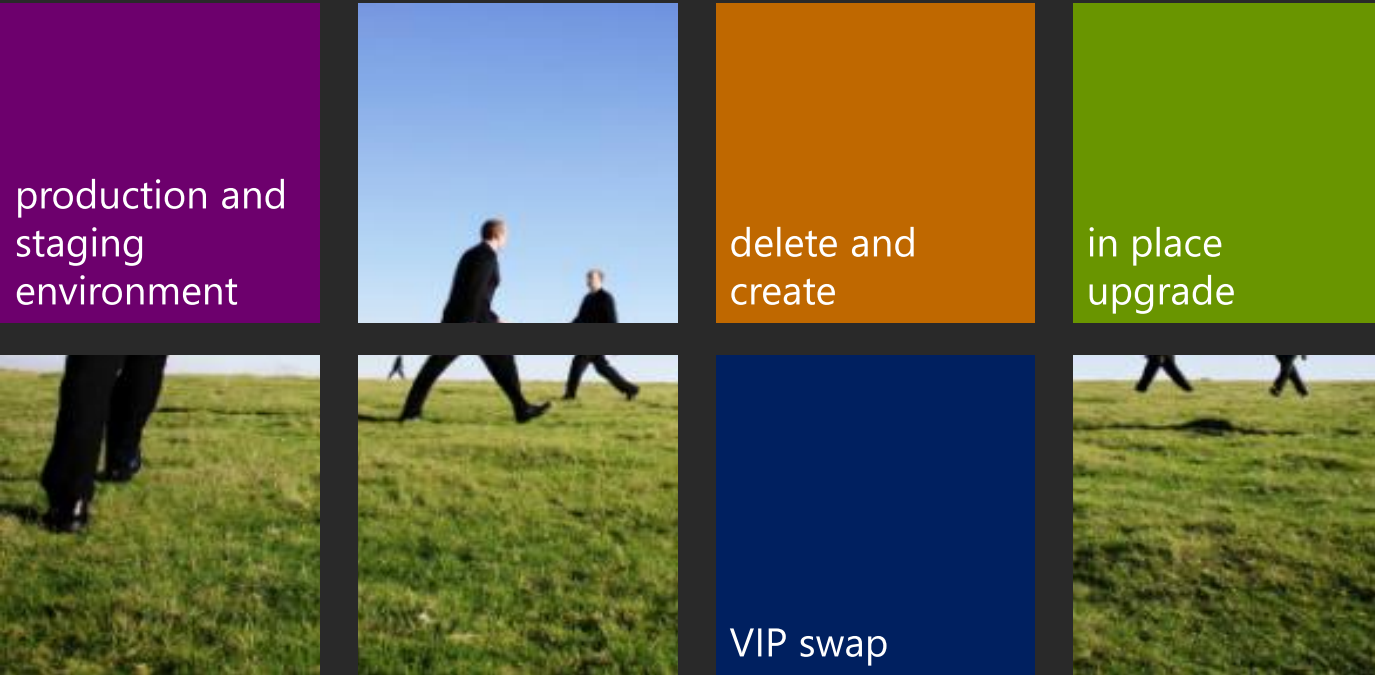
devops



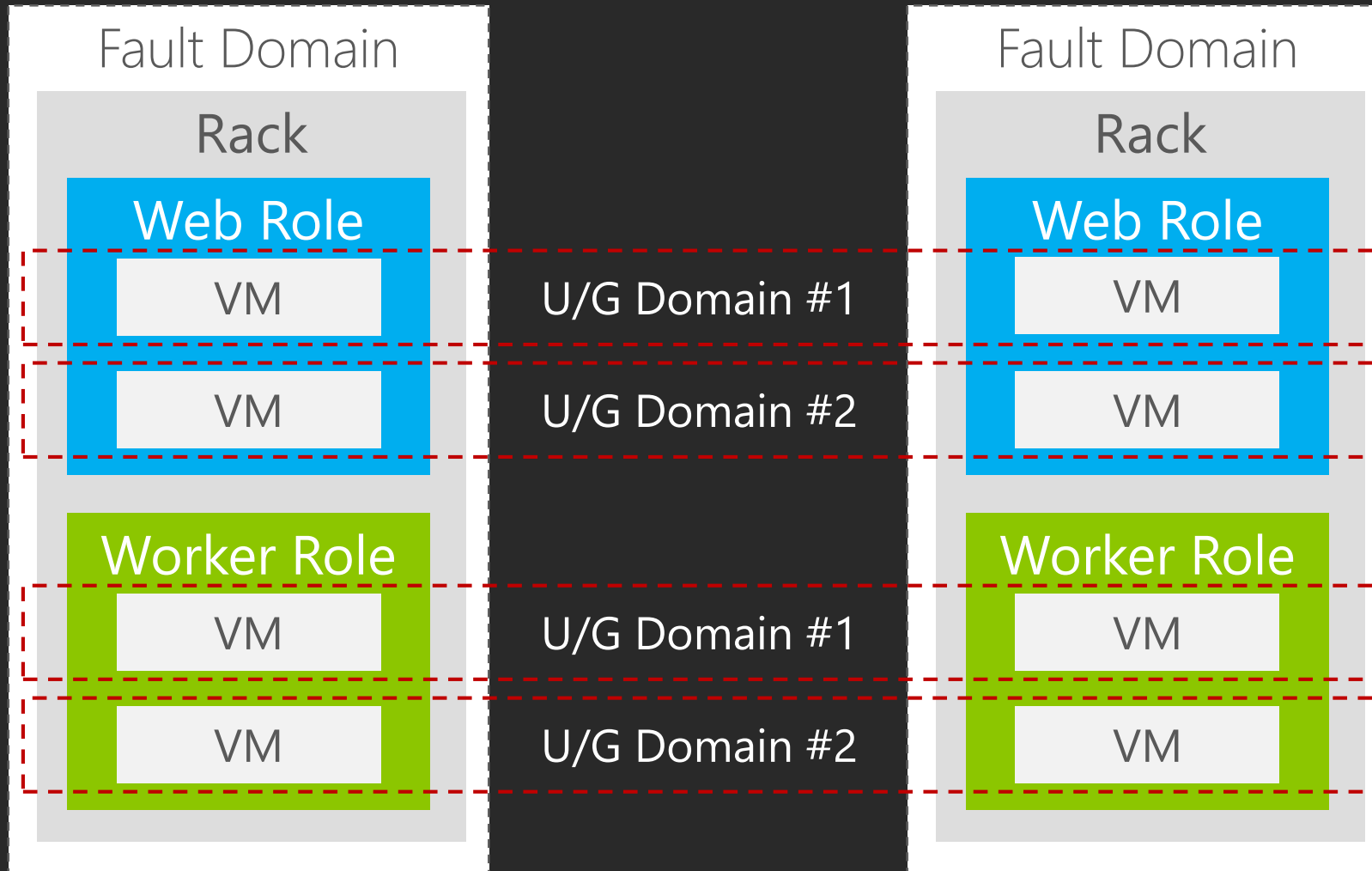
deployment/release strategies



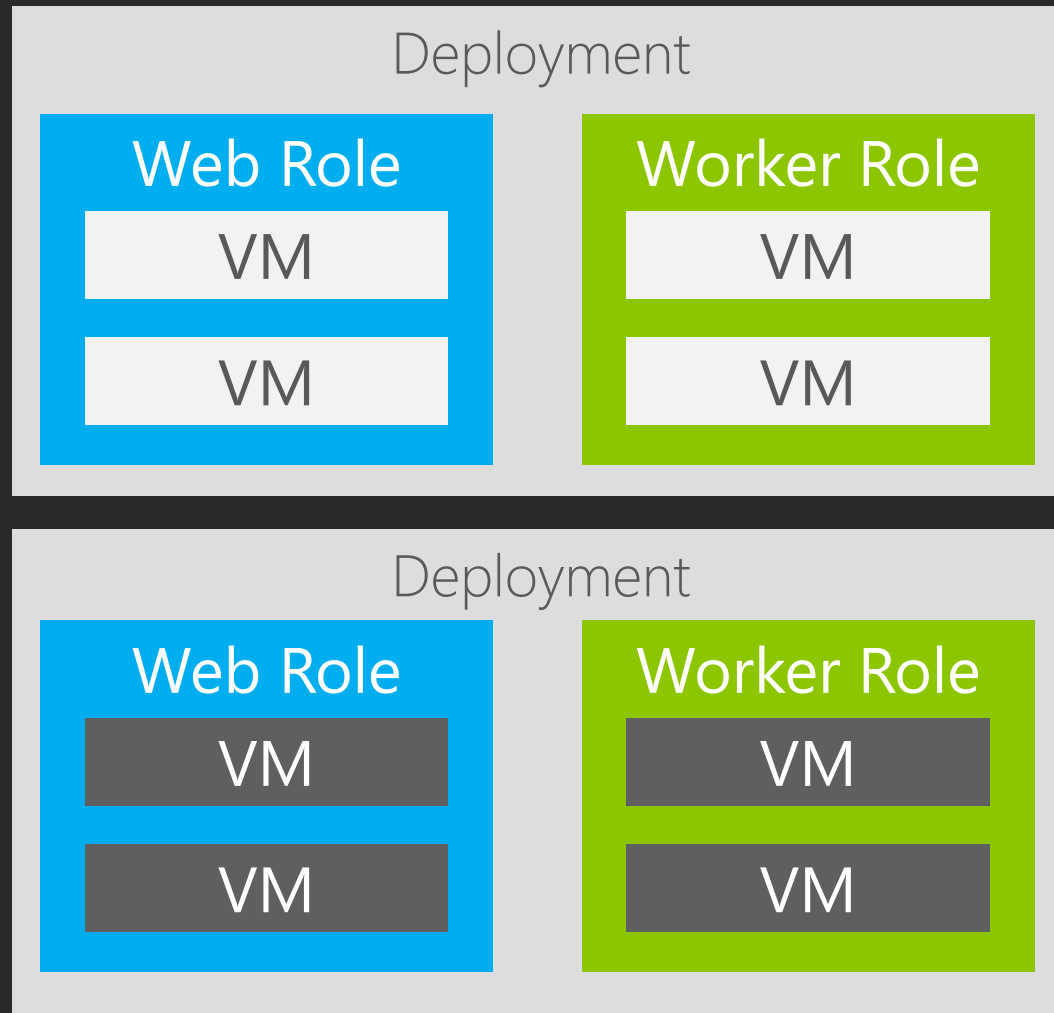
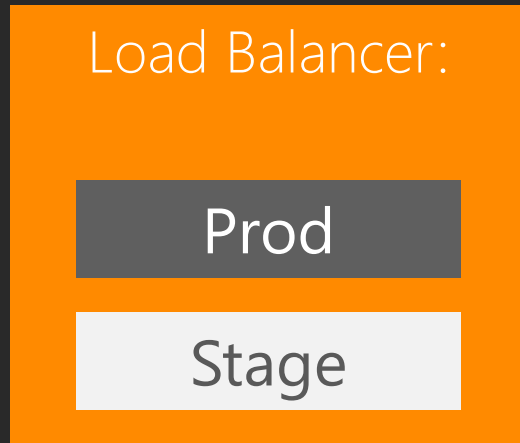
upgrades in Windows Azure



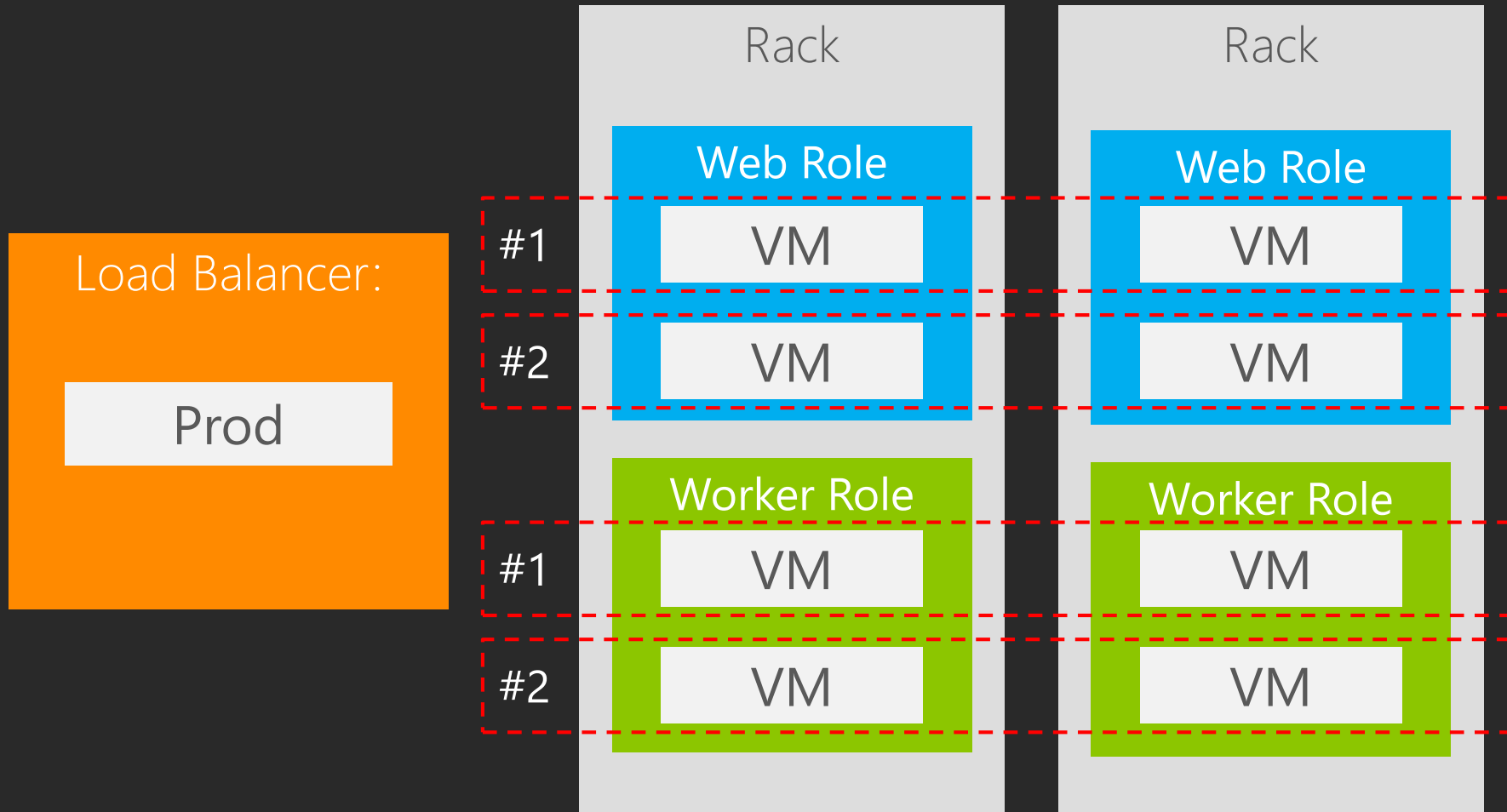
fault and upgrade domains



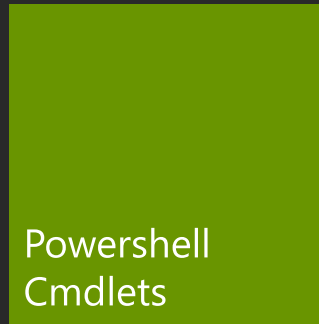
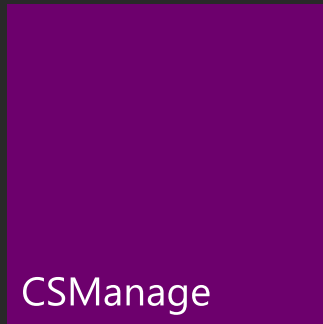
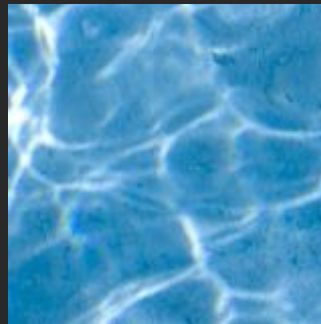
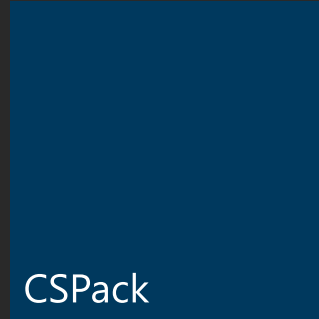
VIP swap

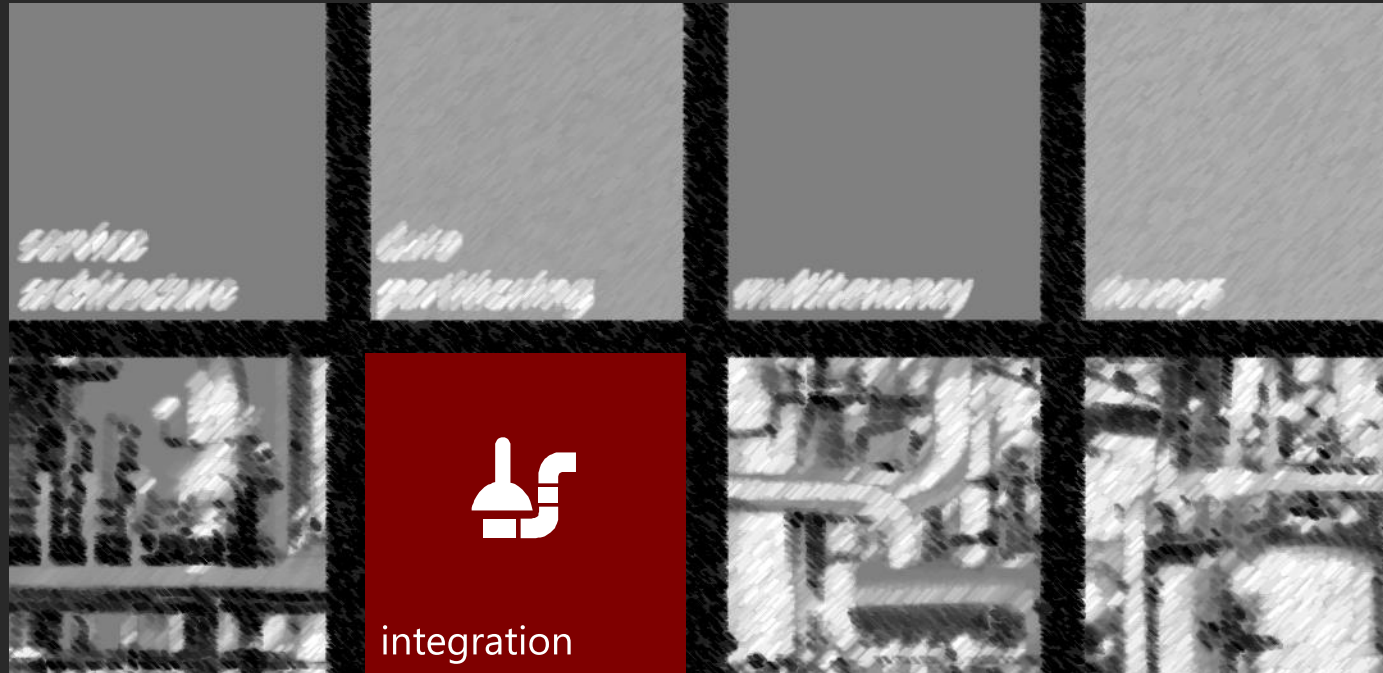


in place upgrade




Windows Azure deployment tools

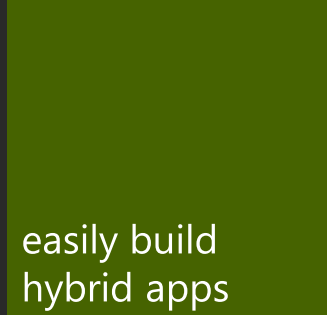




Windows Azure service bus



secure
messaging and
relay capabilities



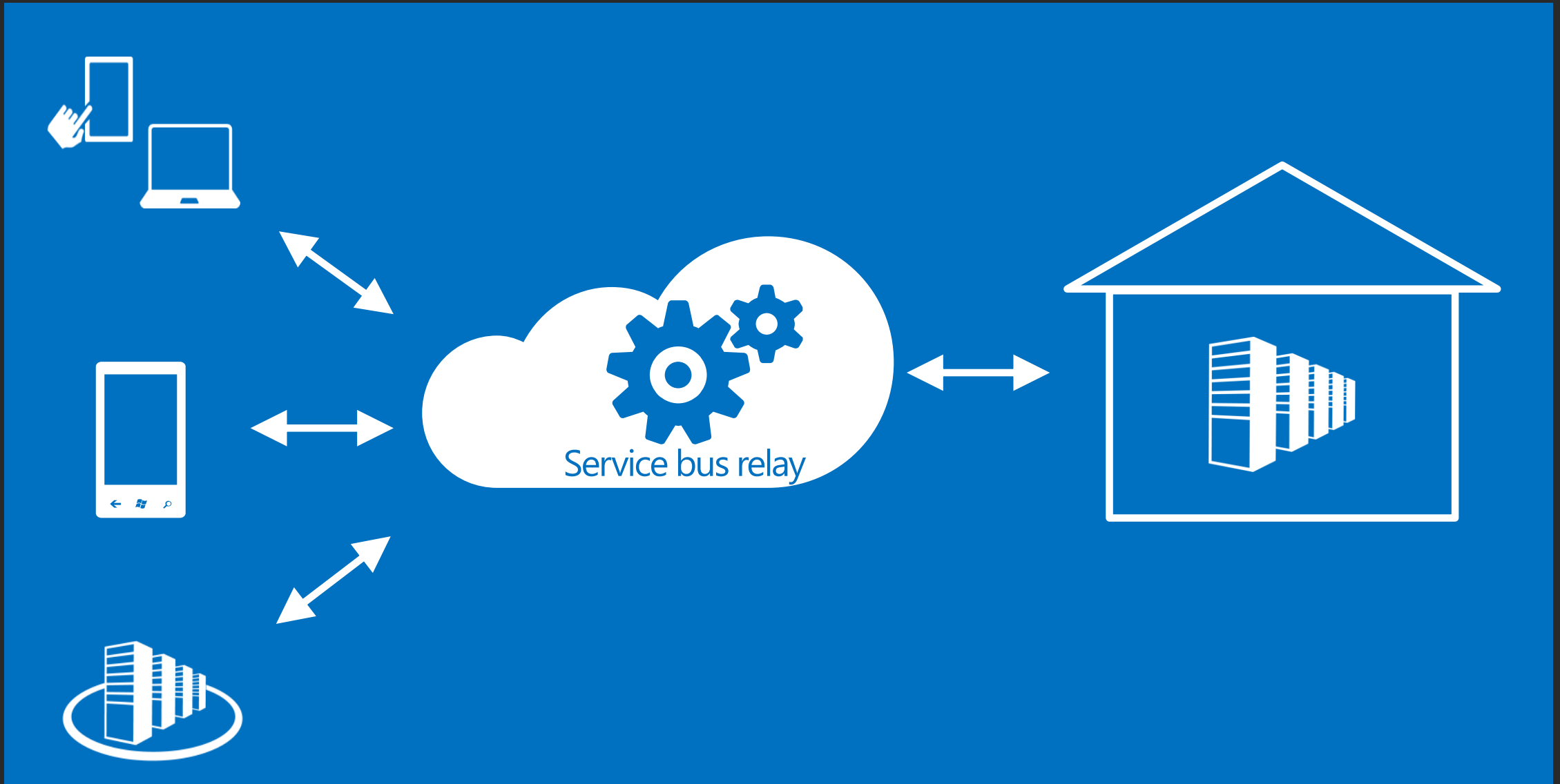
easily build
hybrid apps



enable loosely
coupled
solutions



accessing on-premise services





Demo:
Service Bus
Relay

conclusion



cloud style computing designs for scale out

scale out requires partitioning



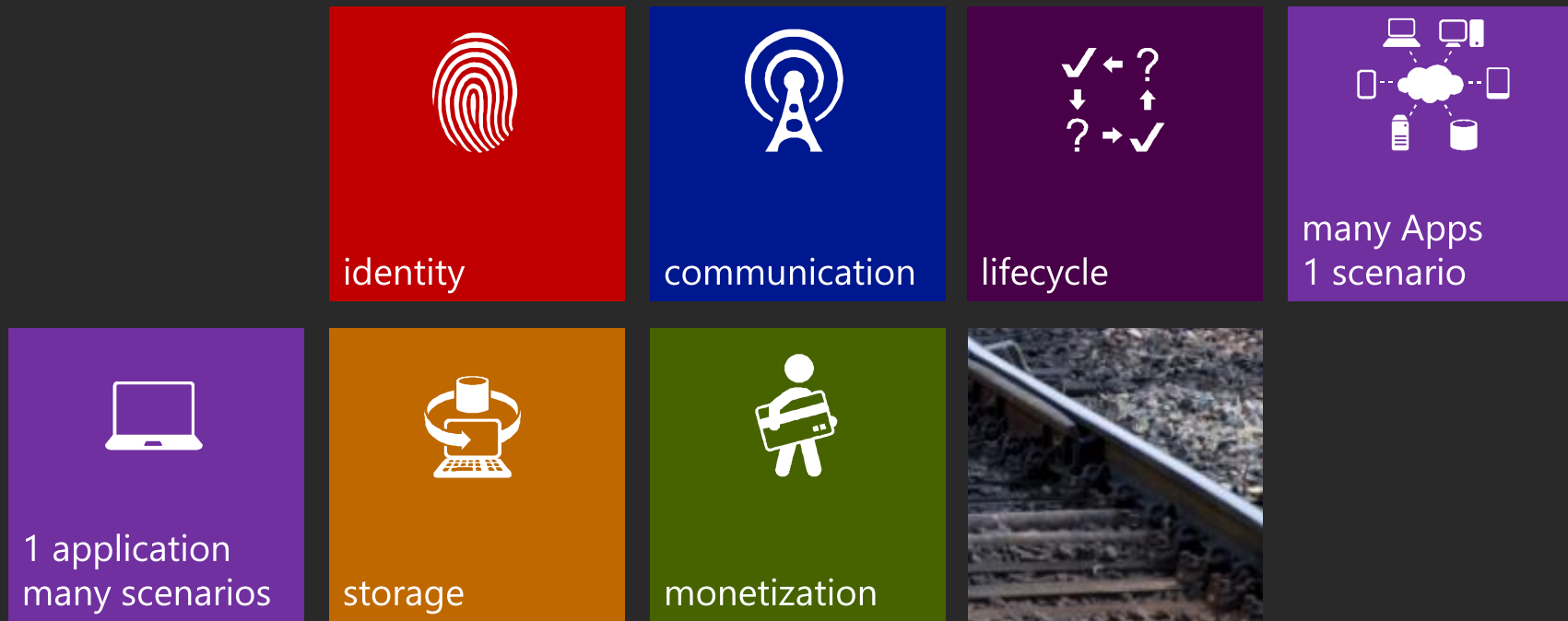
having redundant/duplicated data is ok

continuous delivery is a key asset



chapter V

from applications to apps



conclusion



from applications towards Apps

scenarios and tasks span multiple devices



cloud is a key enabler for connected devices

Windows Azure supports all devices

