Microsoft

Service Fabric - Use Case Scenarios and Examples

Microsoft Services

Conditions and Terms of Use

Copyright and Trademarks

How to View This Presentation

- All slides have two styles of PDF documents:
  - Full View – full page view per slide
  - Notes View – 3 slides per page with ruled lines used to take notes
- Use Adobe Reader or other PDF viewing tools to review the PDF slides
- All PDF slides are located in your AzureSFWS.zip file

Microsoft

Service Fabric - Use Case Scenarios and Examples

Applying Microservices on the Microsoft Platform

Microsoft Services

300+ Service Fabric Customers

Care Otter

neworbit
more than software

janison

XOMNI

semasio

CARBONITE

cardinal
PEOPLE.IDEAS.SOLUTIONS

iGOTCHA

InfoSupport
Solid Innovator

ProsessPilotene

OSIsoft.

QUORUM
BUSINESS SOLUTIONS, INC.

MESH
SYSTEMS

SEKOIA

ais

fuse

Orckestra

Jitneytrade

NeoGeo
New Media GmbH

nimbleschedule.™

TalkTalk
Brighter Phone & Broadband

SolidQ

Reply
solidsoft

AMIDO

AFAS
software

pnop

SO/AT

BALZANO
MICROSOFT CLOUD PROFESSIONALS

Fidelity
INVESTMENTS

K2

UX
RISK

N3

mobit

readify

Triple IT

FULL SCALE 180

kyrt

P97
networks

Omny

UNiFY
SOLUTIONS

CHEVALPARTNERS

Xpirit
Think ahead. Act now.

{x}

Schneider
Electric

LOFTYSOFT

P2
ware

opulent | asp
www.opulentasp.com

UNIT4
In business for people.

solliance
your solution experts

Genetec

Runpath

Confused.com

PENDRICA

ILLYRIAD
GAMES

grass valley
A BELDEN BRAND

daenet

codit
integrating your business

scoreBIG
Save on every ticket. Guaranteed.

genisoft
intelligence in business

nexum

OpenTee

OFFBEAT
SOFTWARE SOLUTIONS PVT. LTD.

Motion10

WujiGrid

Agenda

- TalkTalk
- Mesh Systems
- Illyriad Games

Service Fabric - Use Case Scenarios and Examples

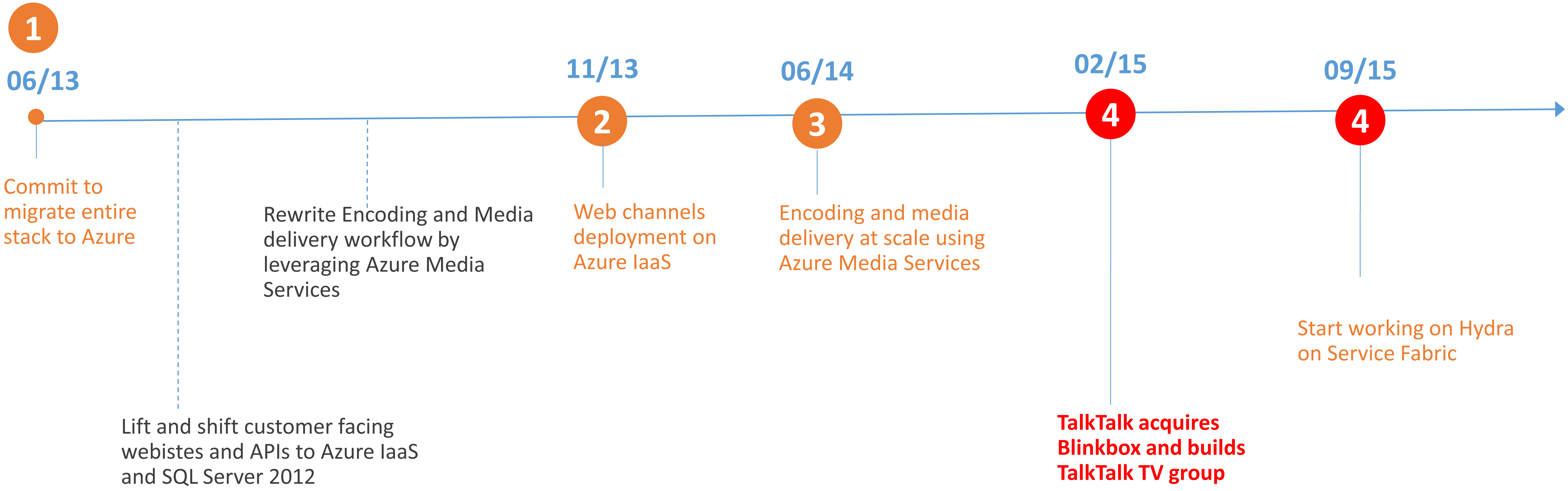
TalkTalk TV

Microsoft Services

TalkTalk

- UK’s third largest cable TV Provider
- The Leading value for money phone, broadband and TV Provider for UK homes.
- With more than 1.4M customers, TalkTalk is the # 3 pay TV platform in the UK
- TalkTalk TV delivers the latest TV and movie content to TalkTalk subscribers, Apple computers, PCs, game consoles, tablets, and smart TVs.

TalkTalk TV – A long Journey to the Cloud



1

06/13

Commit to migrate entire stack to Azure

Lift and shift customer facing webistes and APIs to Azure IaaS and SQL Server 2012

Rewrite Encoding and Media delivery workflow by leveraging Azure Media Services

11/13

2

Web channels deployment on Azure IaaS

06/14

3

Encoding and media delivery at scale using Azure Media Services

02/15

4

TalkTalk acquires Blinkbox and builds TalkTalk TV group

09/15

4

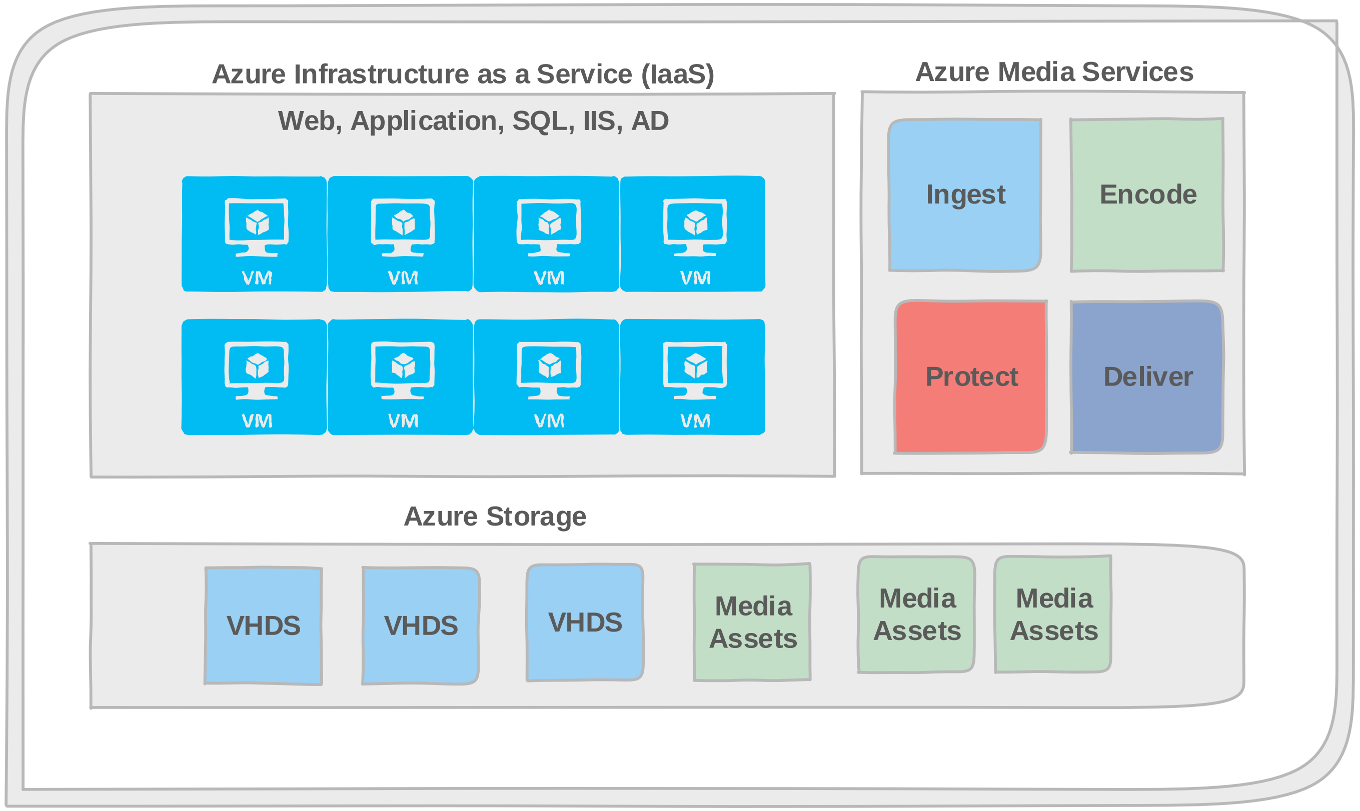
Start working on Hydra on Service Fabric

Hydra

Single Encoding Pipeline

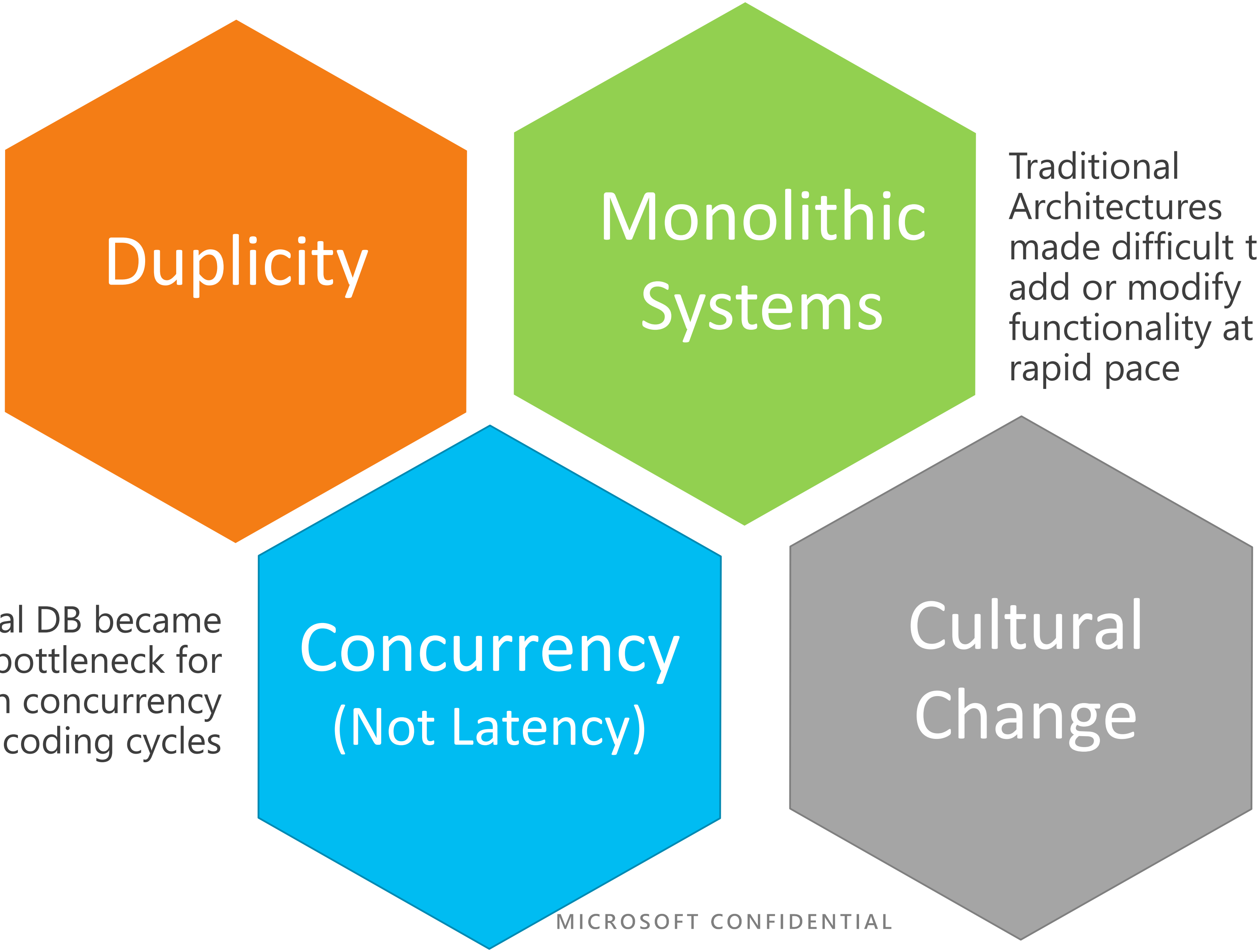
oatly – Lead Engineer, TalkTalk TV

TalkTalk TV As IaaS



- > 200 virtual machine cores
- > 1.5 PB of Azure Storage
- > Media assets delivered from Azure NEU and WEU
- > Hybrid SQL Cluster (2 copies in the Cloud and an additional on-premises)

TalkTalk TV Challenges



Duplicity

TalkTalk And Blinkbox had similar backend systems

Monolithic Systems

Traditional Architectures made difficult to add or modify functionality at a rapid pace

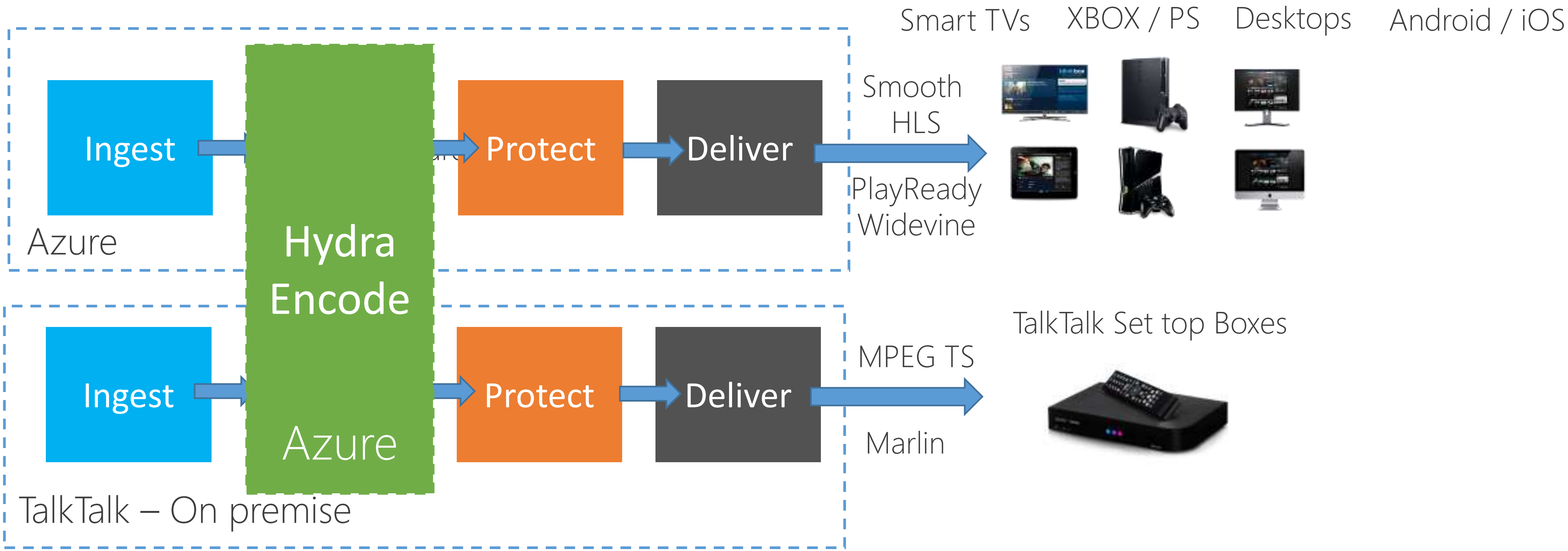
Concurrency (Not Latency)

Central DB became a bottleneck for high concurrency encoding cycles

Cultural Change

TalkTalk TV Leadership wanted to embrace more autonomous and experimental paradigms.

TalkTalk TV Challenges - Duplicity



Smart TVs

XBOX / PS

Desktops

Android / iOS

Ingest

Hydra Encode

Protect

Deliver

Azure

Smooth HLS

PlayReady Widevine

TalkTalk Set top Boxes

Ingest

Azure

Protect

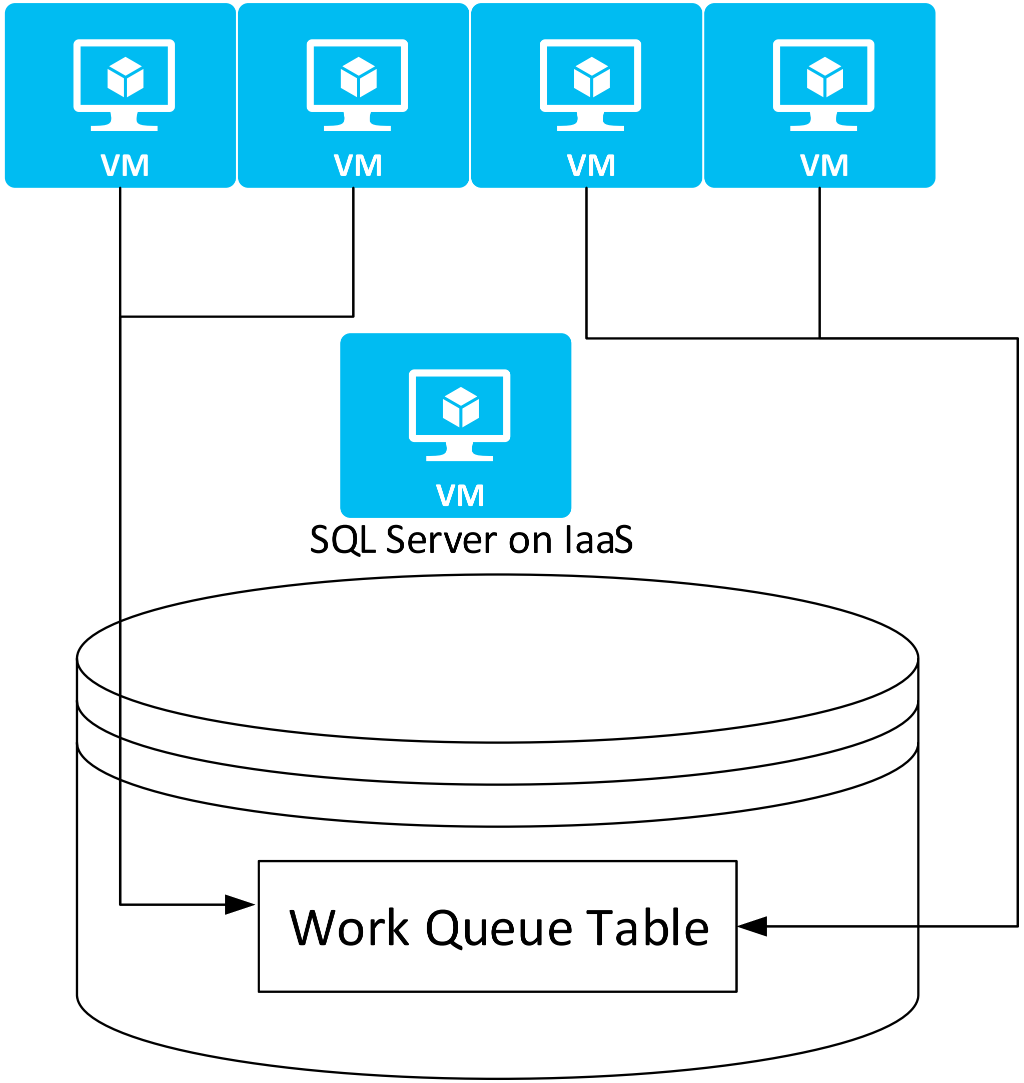
Deliver

MPEG TS

Marlin

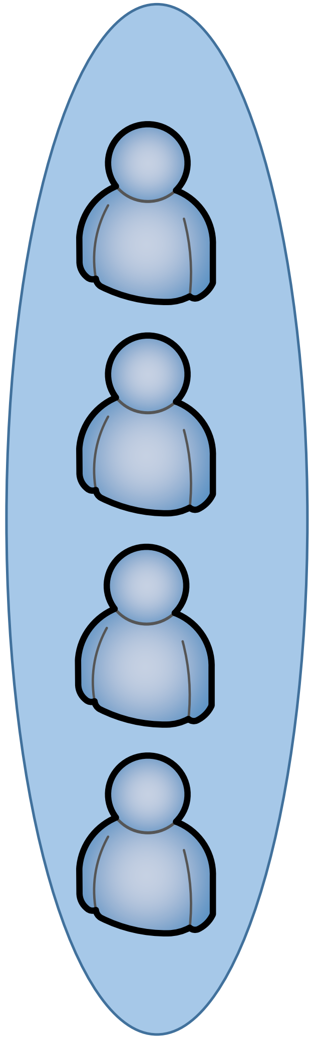
TalkTalk – On premise

Monolithic Systems / Concurrency



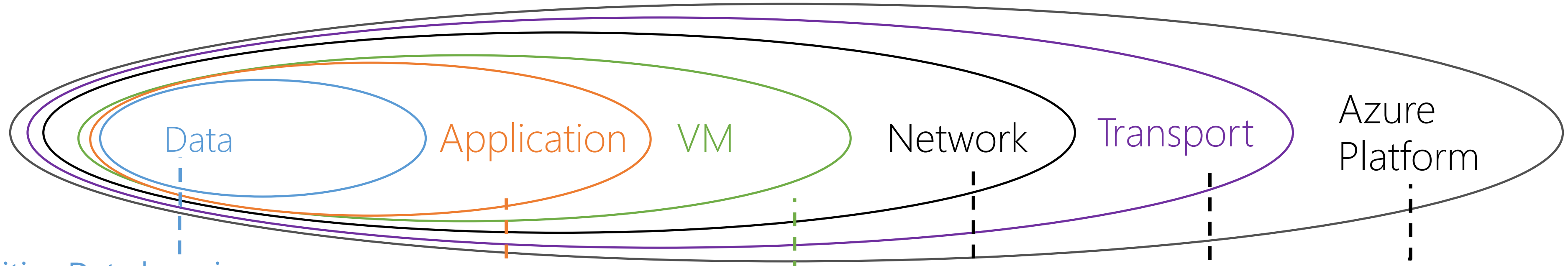
- During large batches of encoding cycles the DB suffered high contention
- Despite having an elastic platform (Azure Media Encoders), the ability to submit large encoding jobs was constrained by traditional design paradigms
- Too tightly coupled APIs / systems

Cultural change to development



- Needed to adopt a more nimble and autonomous mentality
- Build small and cross functional teams
- Hydra : 3 engineers + Microsoft Support
- Developers + IT Ops – how to interact

Adopting an overall approach to security



Data

Application

VM

Network

Transport

Azure Platform

Encrypt Sensitive Data by using AES Encryption and Azure Key Vault

or

Use external Data sources (Azure SQL DB)

Secure Application API Endpoints by using Azure Active Directory and OWIN authentication

Secure Service Fabric Management Endpoints

Don’t Expose RDP or Any Other Endpoint Directly

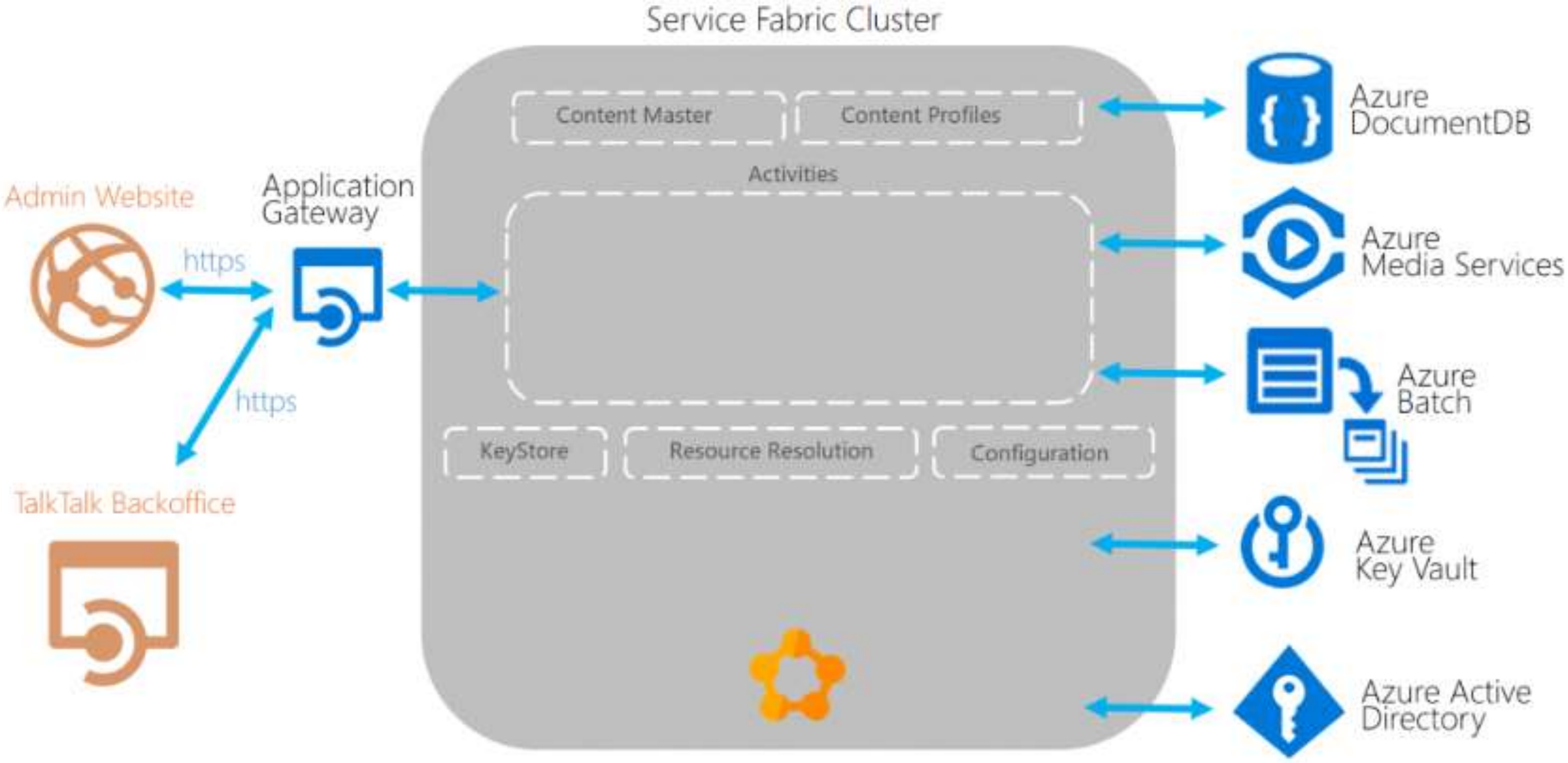
Use Jumpboxes and LBs

Apply Network Security Groups
and
Web Application Firewalls

HTTPS - SSL Termination Managed By Azure Application Gateway

Applied Transparently to us

Hydra on Azure Service Fabric – Conceptual Design



Hydra on Service Fabric – Authentication and Security

Azure AD - Applications

- Azure AD for user authentication to SF Apps
- Certificates/Secrets for cross-service authentication

Azure Key Vault

- Store Azure keys (Media Services, Batch, DocumentDB…)
- Configuration settings
- Certificates

Hydra on Azure Service Fabric – Media Jobs

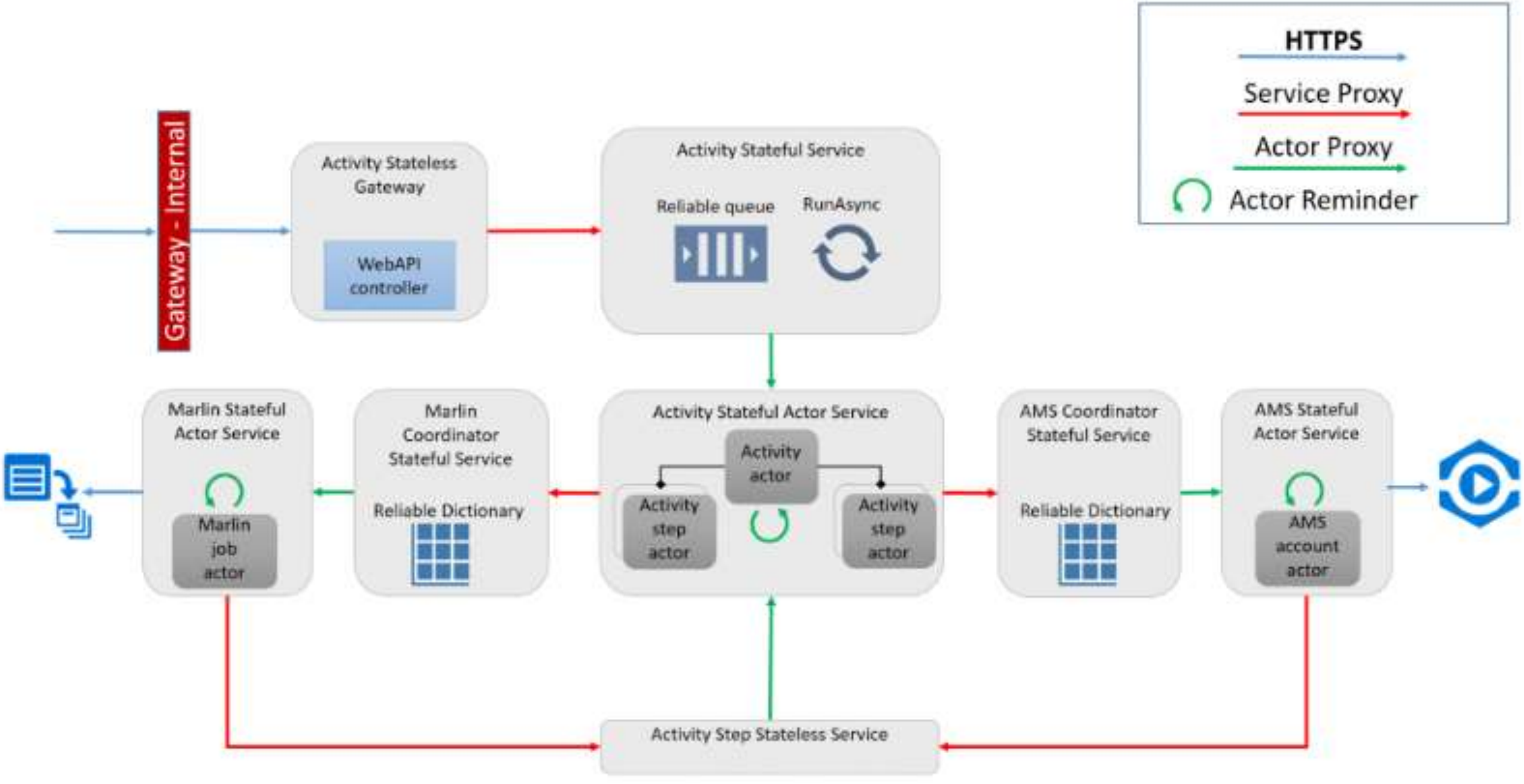
Media Services

- Azure encoders provide elastic encoding
- Azure Streaming Services to deliver the content to IP based devices

Azure Batch

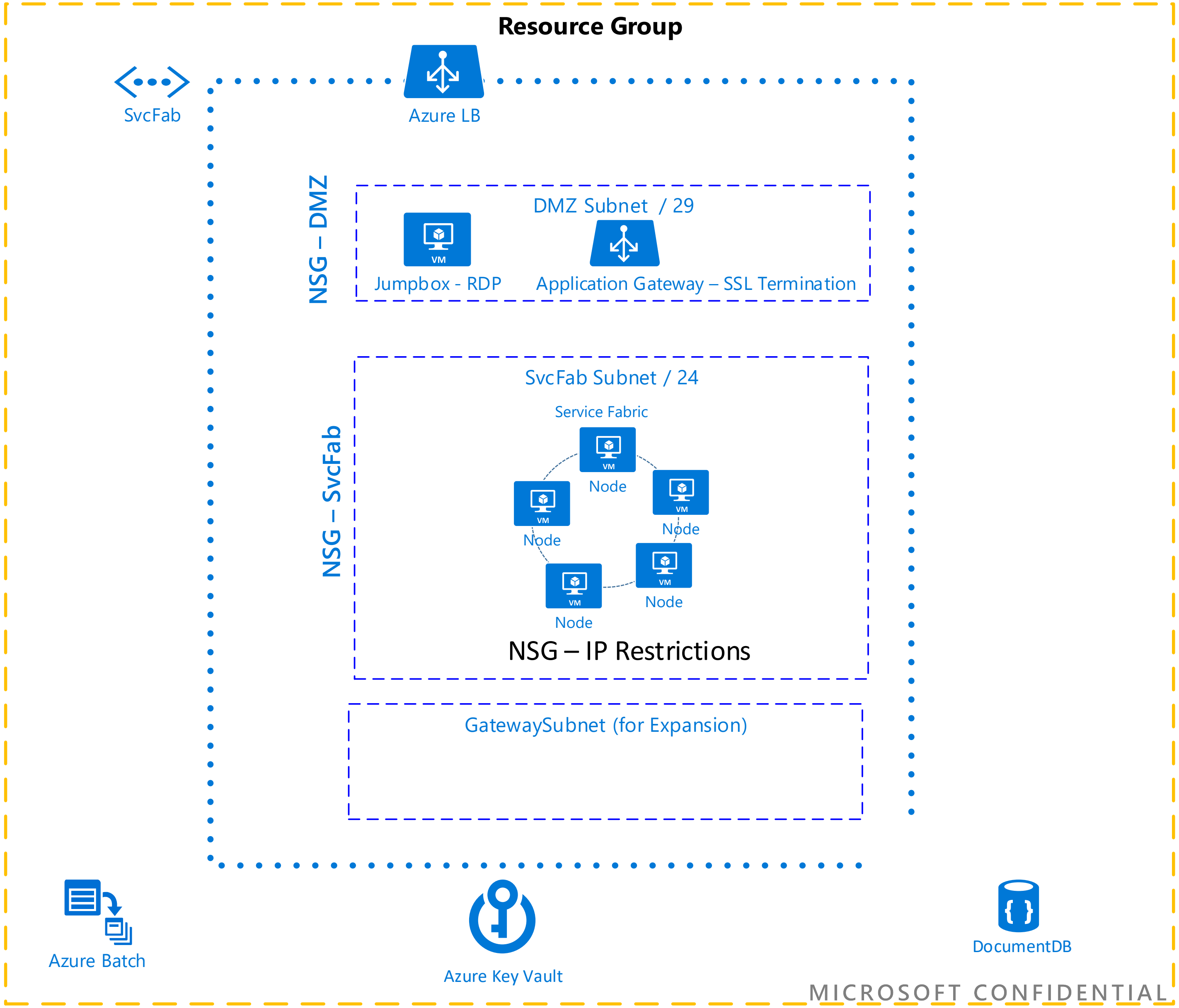
- Extends Azure Media Services capabilities
- Used for Marlin Encryption and long running jobs

Hydra on Azure Service Fabric – Detailed Services Design



HTTPS
Service Proxy
Actor Proxy
Actor Reminder
Gateway - Internal
Activity Stateless Gateway
WebAPI controller
Activity Stateful Service
Reliable queue
RunAsync
Marlin Stateful Actor Service
Marlin job actor
Marlin Coordinator Stateful Service
Reliable Dictionary
Activity Stateful Actor Service
Activity actor
Activity step actor
Activity step actor
AMS Coordinator Stateful Service
Reliable Dictionary
AMS Stateful Actor Service
AMS account actor
Activity Step Stateless Service

Hydra on Azure Service Fabric – Infrastructure Design



- Use Azure resource manager templates for deployment
- Application of security policies :
  - SSL termination
  - Endpoints authentication
  - IP restrictions

Lessons Learned from TalkTalk

- Brown field development
  - You do not have to re-write the whole app to use microservices approach
- Pay attention to application and service composition
  - Loose coupling and high cohesion (contracts)
- Use the right programming model for each scenario
- Determine security and telemetry requirements up front
- Small team agility - The Perfect Combination
  - A Small Team (3 “+1”) with a nimble engineering mentality
  - Right platform (Azure/ Service Fabric /Media Services/Batch)

Microsoft

Service Fabric - Use Case Scenarios and Examples

IoT on Service Fabric : Mesh Systems

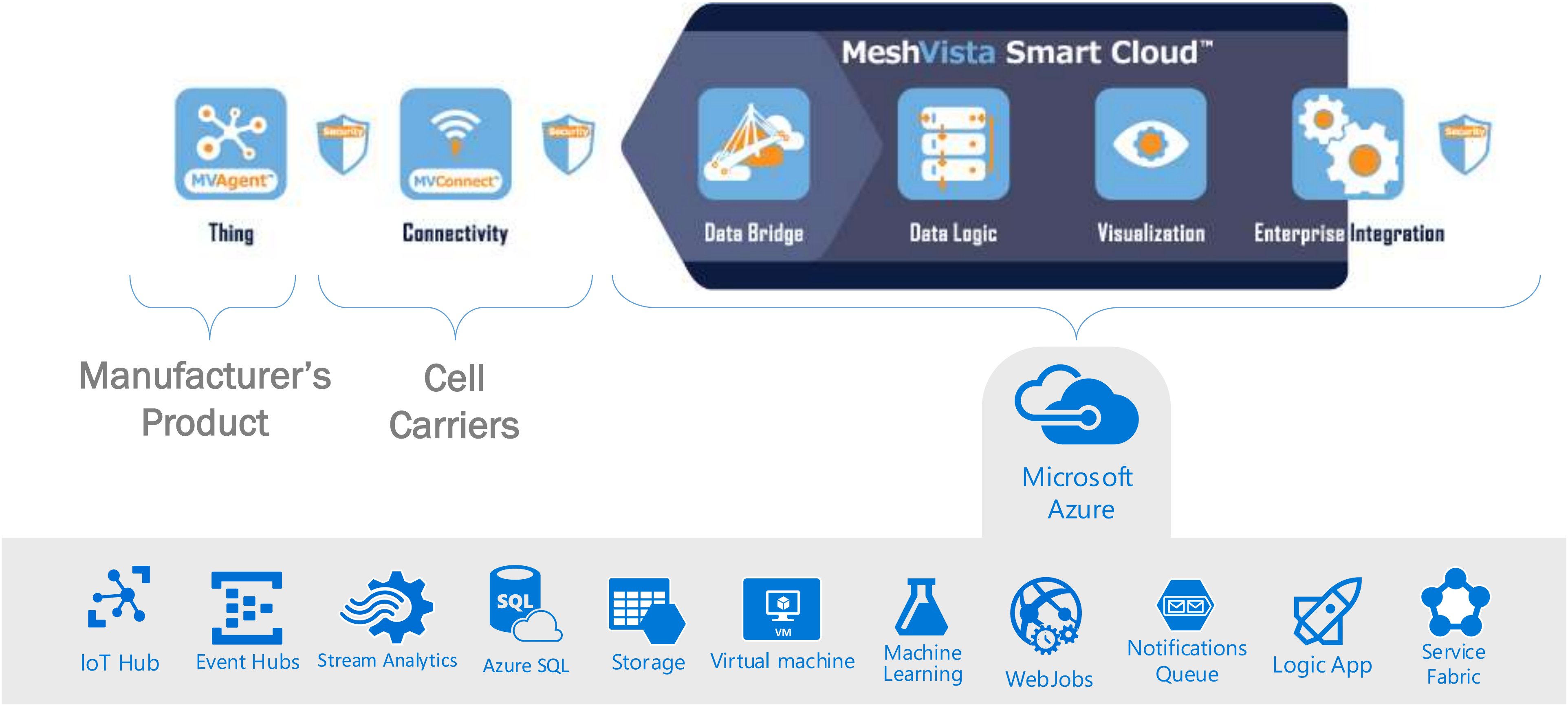
Microsoft Services

IoT On Azure Service Fabric

- Mesh Systems – NetLink
- Goals Of project
- Architecture
- Performance Metrics
- Lessons Learned

Mesh Systems

- IoT Software and Services
- Connect Devices To Our Platform in Azure



MeshVista Smart Cloud™

Thing

Connectivity

Data Bridge

Data Logic

Visualization

Enterprise Integration

Manufacturer's Product

Cell Carriers

Microsoft Azure

IoT Hub

Event Hubs

Stream Analytics

Azure SQL

Storage

Virtual machine

Machine Learning

WebJobs

Notifications Queue

Logic App

Service Fabric

Smart Energy

Product Manufacturer: WLS LIGHTING SYSTEMS

Business Case: Reduce maintenance and energy costs by only running lights when and where necessary.

Device: Multi-channel Wireless Light Controls and Base Station

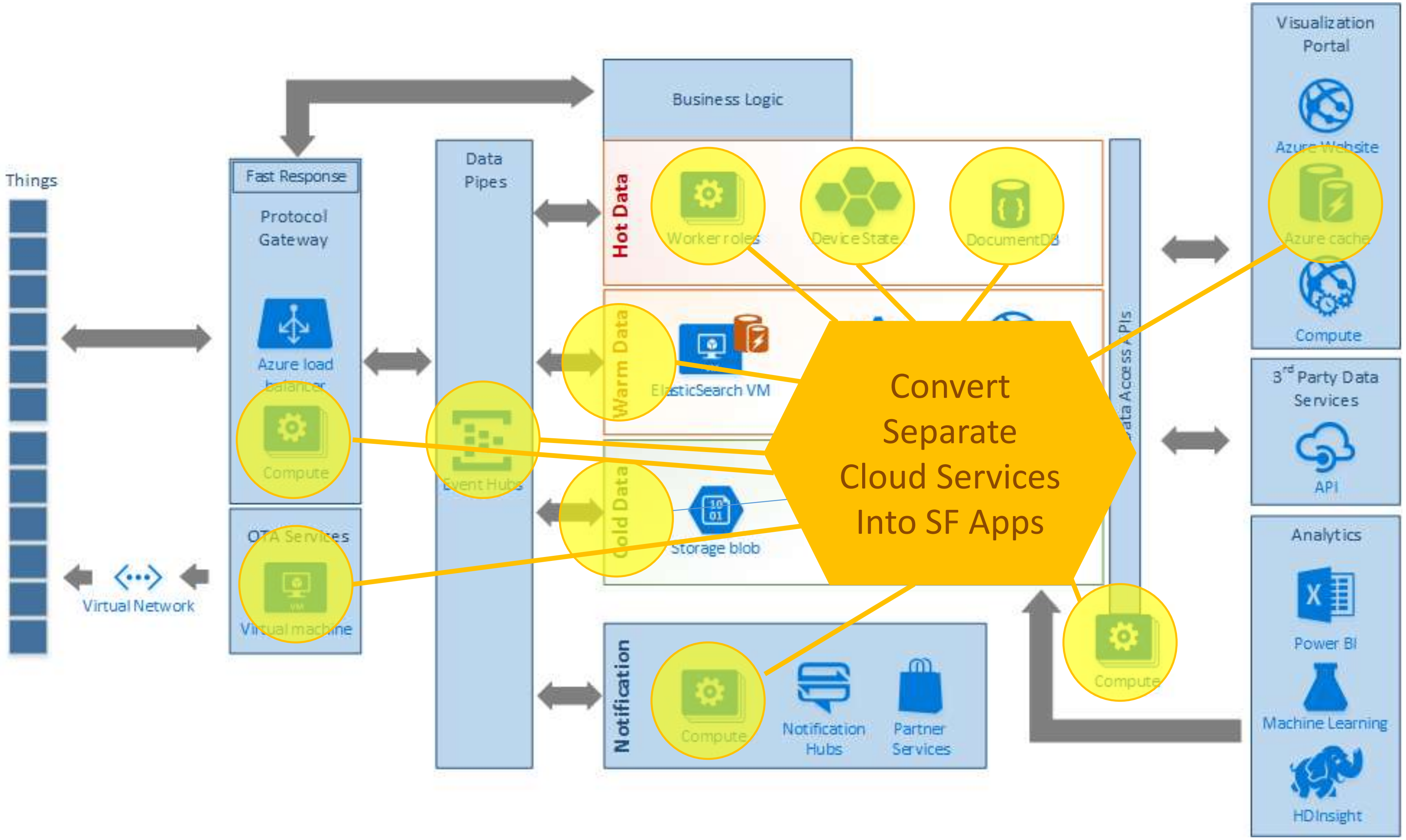
Azure : Data logging, remote scheduling, event notifications, energy usage analysis

Key Comments:

- Clients see 12-19 month ROI
- Improved safety and quality of service

PaaS V1 Architecture

MESHVista®



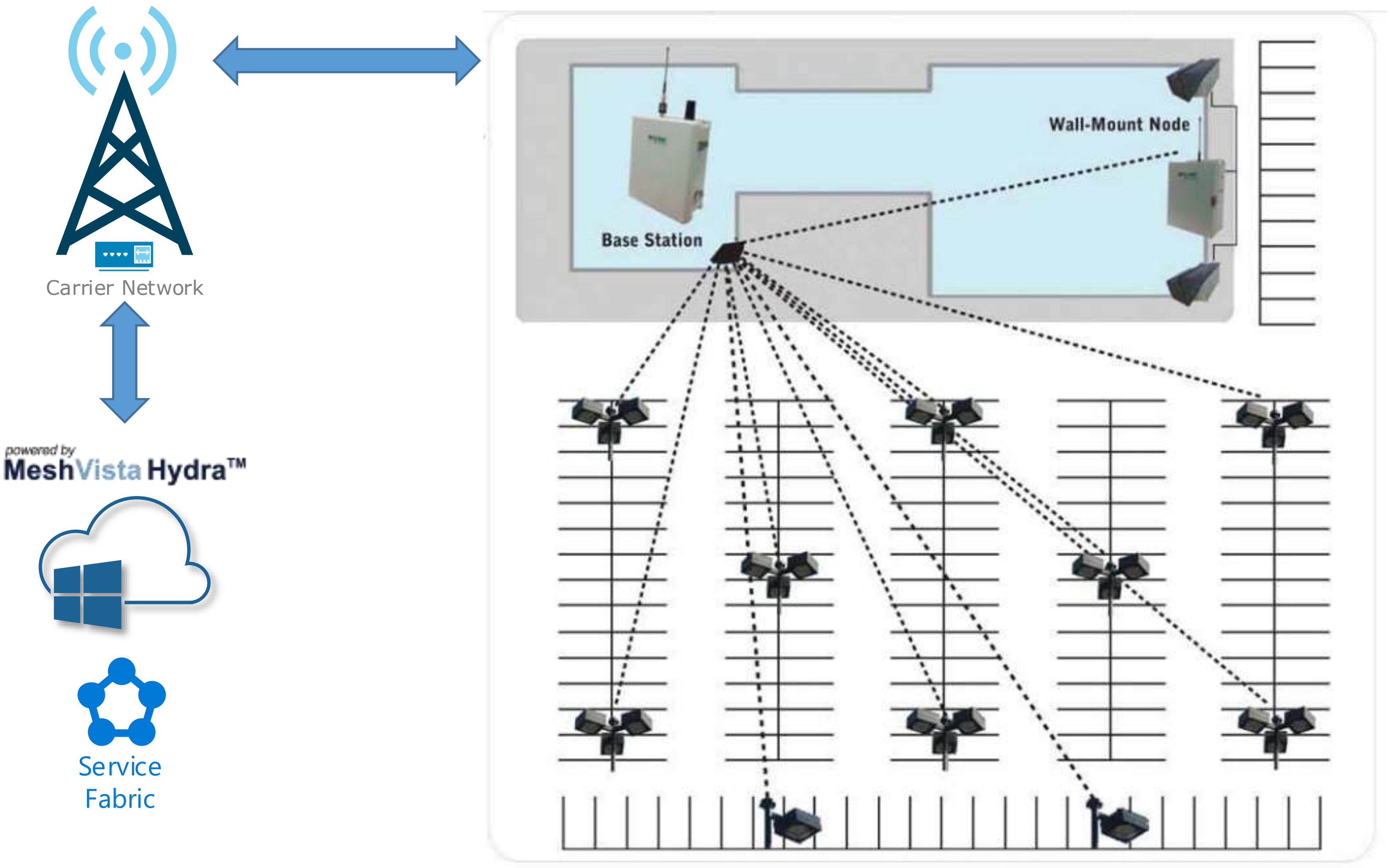
Convert Separate Cloud Services Into SF Apps

Goals of project

- Bare-Metal Migration to Azure
  - Aging Infrastructure needs to be replaced
  - No easy way to scale with old model

- Why Choose PaaS V2 vs V1?
  - Reduce cost and improve resilience
  - 3 Worker/Web Roles for each task to meet SLA
    - Role only running < 10% capacity
  - Device State / Actor Pattern – light fixtures represent Actors, not available in V1

NetLink WAN – Data flow

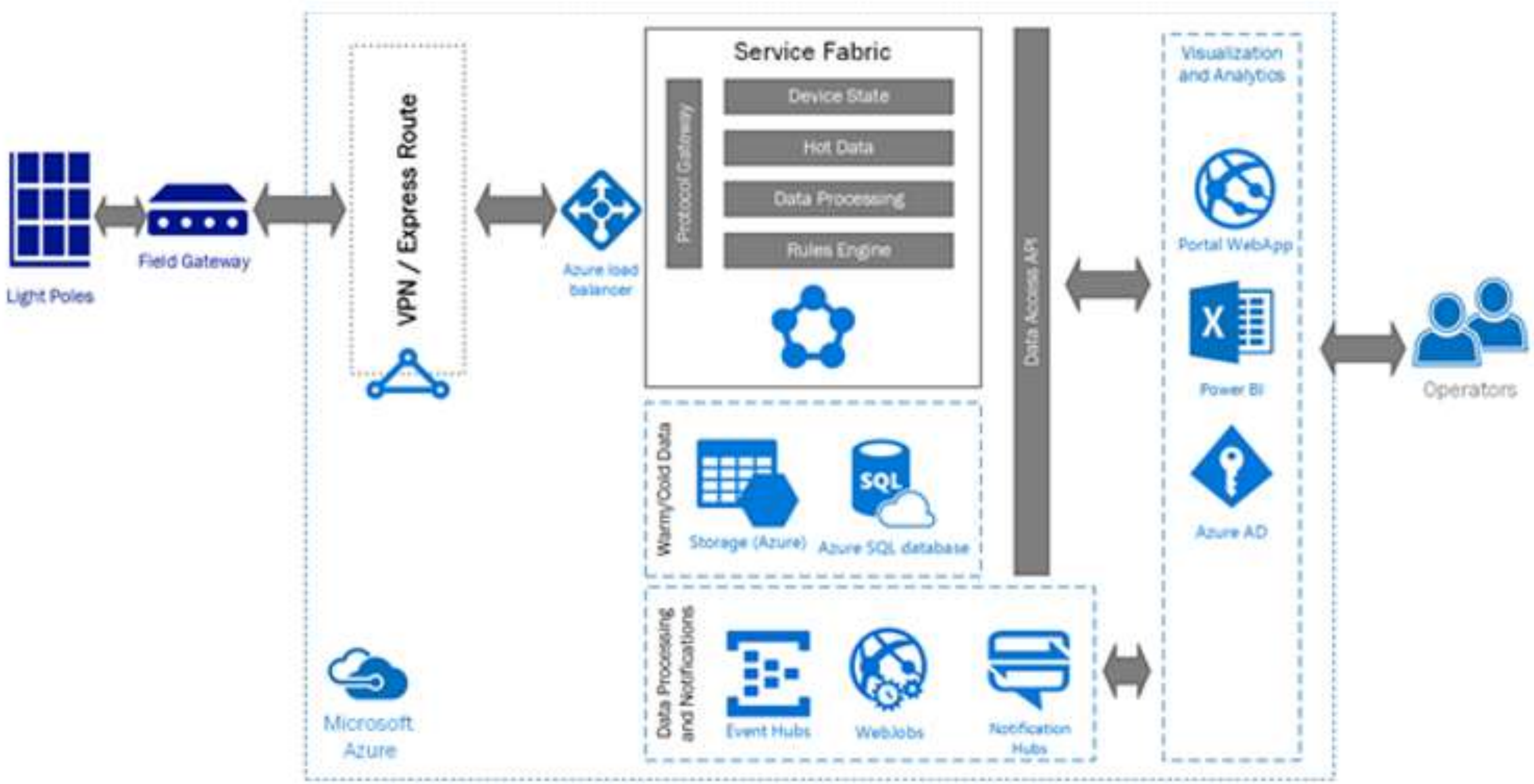


Carrier Network

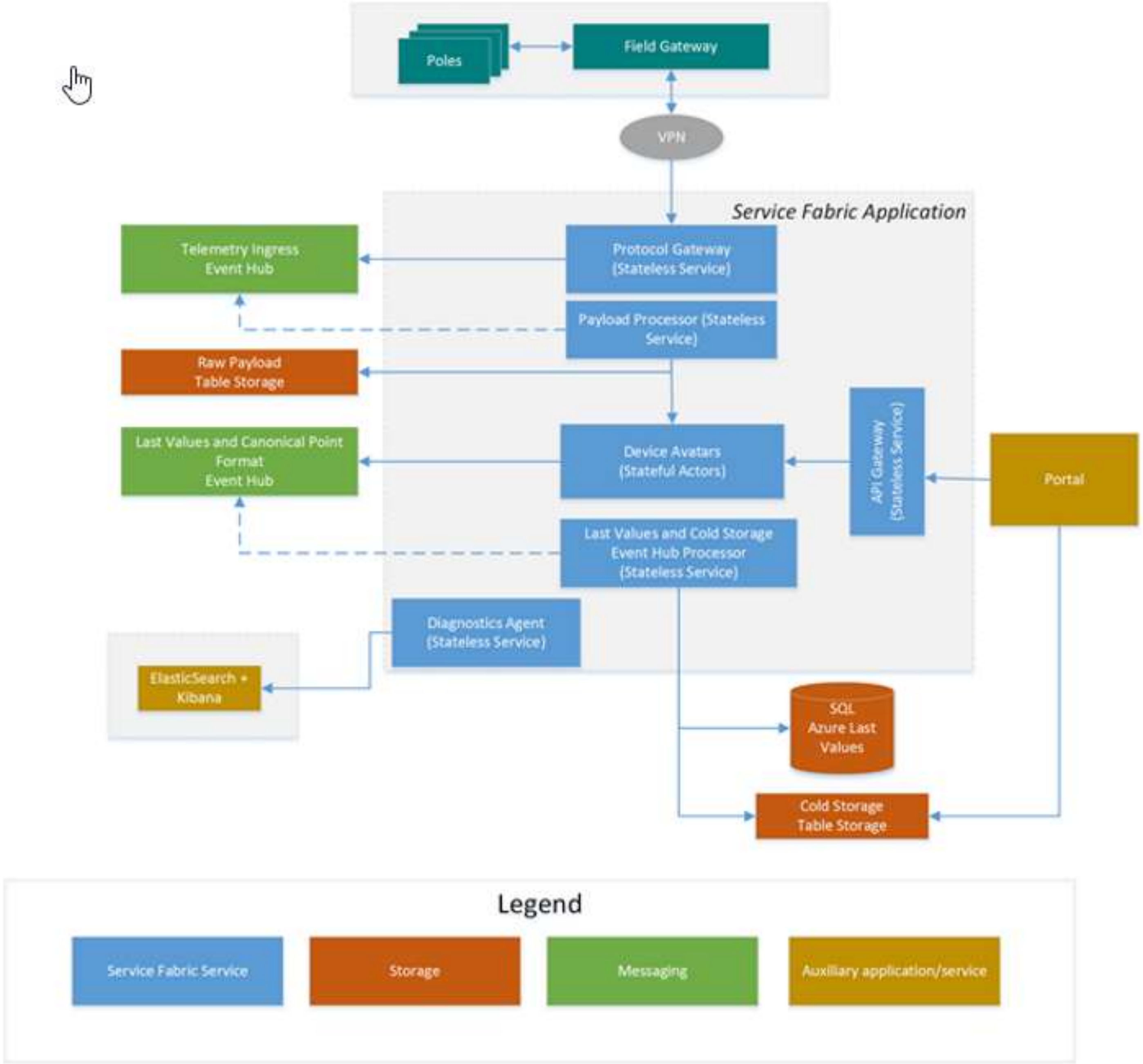
powered by MeshVista Hydra™

Service Fabric

MeshVista Hydra Architecture



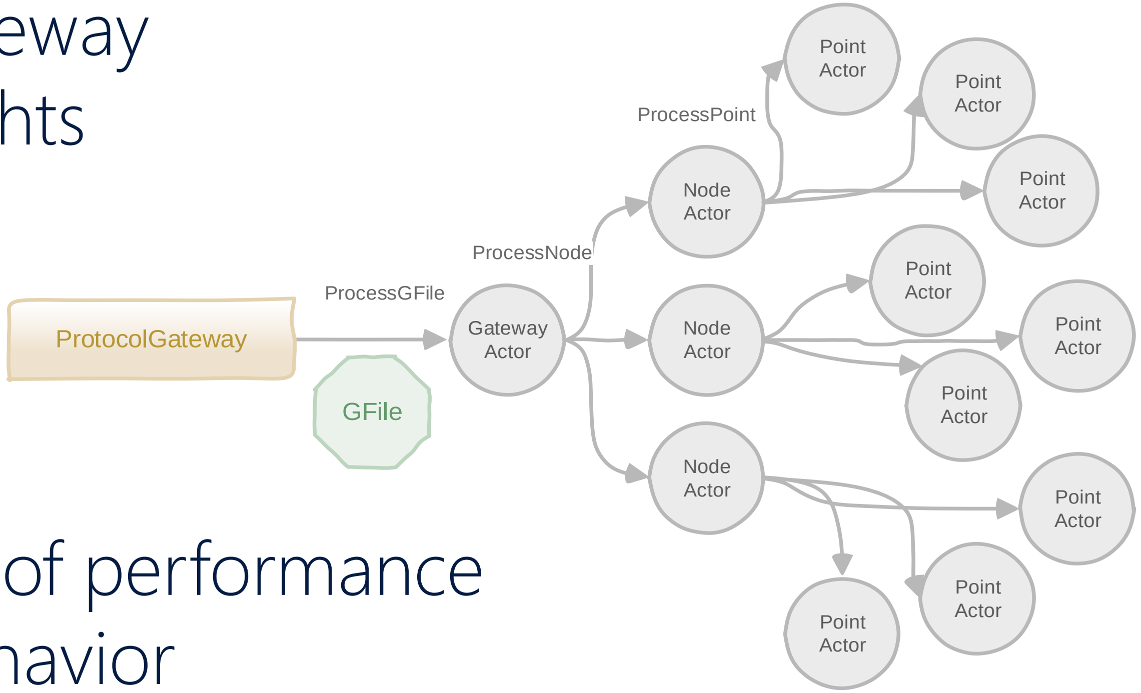
Mesh Systems microservices


Poles
Field Gateway
VPN
Service Fabric Application
Telemetry Ingress Event Hub
Protocol Gateway (Stateless Service)
Payload Processor (Stateless Service)
Raw Payload Table Storage
Last Values and Canonical Point Format Event Hub
Device Avatars (Stateful Actors)
API Gateway (Stateless Service)
Portal
Last Values and Cold Storage Event Hub Processor (Stateless Service)
Diagnostics Agent (Stateless Service)
ElasticSearch + Kibana
SQL Azure Last Values
Cold Storage Table Storage
Legend
Service Fabric Service
Storage
Messaging
Auxiliary application/service

Device Actor Fanout

Each incoming message evokes a fan out of about ~200 actors

Gateway: Field Gateway
Node: Pole with lights
Point: Sensor Value



- Simplicity of actor vs cost of performance
- Business rules of Point behavior (alarms, timers, aggregation rules, etc.)

Environment Applications

Protocol Gateway service – Parses the inbound device payloads

Device Avatars Actors service – Holds device state and runs device specific logic

Cold Storage Application – Persists device time series data to long term storage

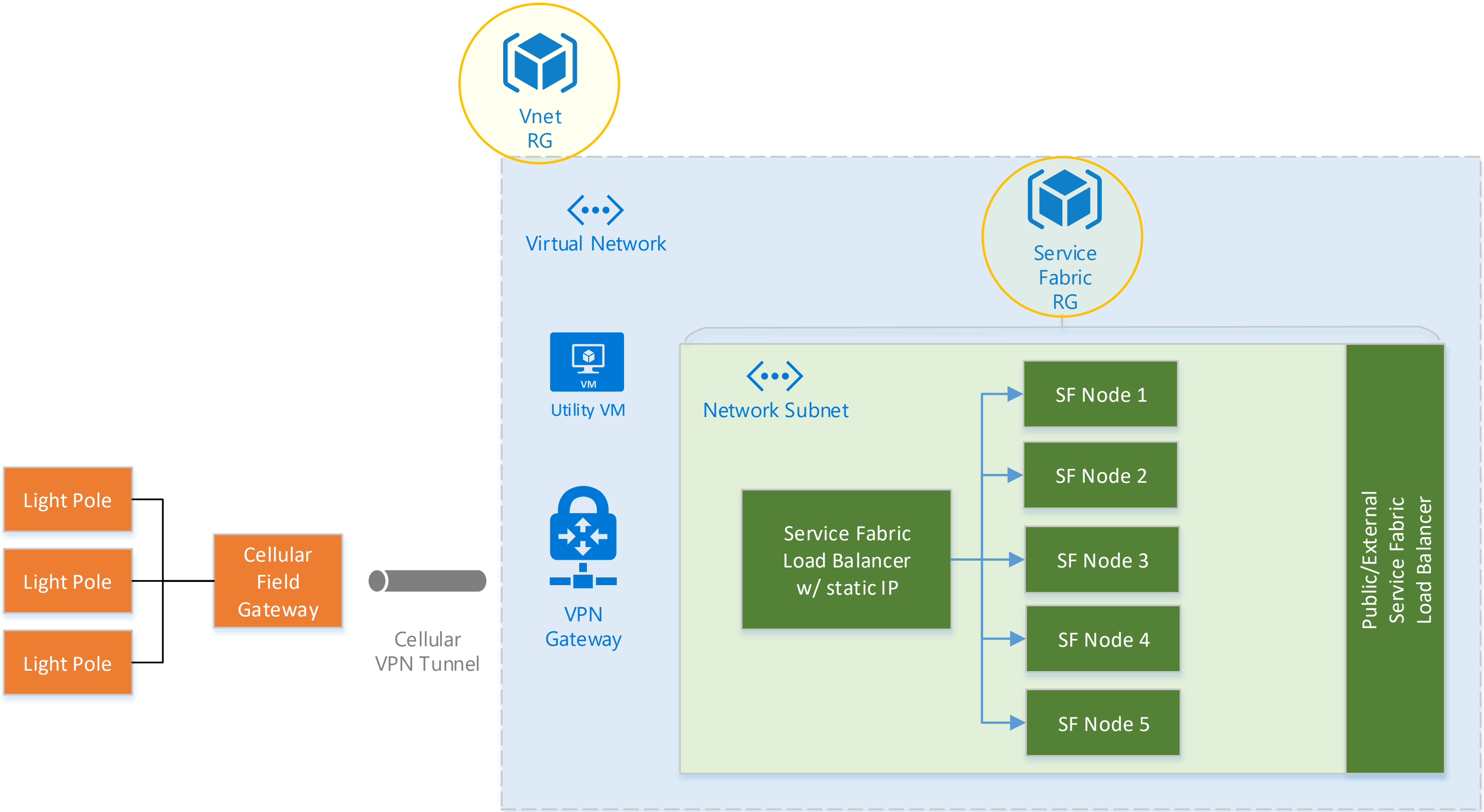
Last Values and Canonical Point Format Event hub service – Persists last value data to query-able storage

Raw Payload Processor service Persists raw, unprocessed device payloads to archive storage

Diagnostics Agent Service – ETW listener forwards app and health diagnostics to ElasticSearch

Elastic Search/Kibana Application – Automated Elastic Search index management

Two ARM Templates for deployment



Performance Metrics

- Class of machine A2 – A3 RAM (5 machines)
  - 300k actors were CPU intensive
  - RAM pegged on A2

- Ways to improve the process
  - Placement Constraints would help
  - Autoscaling Would Help Spikes
  - Azure Customer Advisory Team provide test result…

Details

- Test Scenario
  - 1,500 Gateways, 10 Nodes per Gateway, 20 Points per Node
    - Total: 300,000 actors
  - Gateway receives a GFile every three minutes
    - ~1,650 actors invoked per sec
  - Actor’s State < 150 bytes
  - 10 minutes warm up (activations of all actors)
  - Measured latency of the full fan-out of a Gateway

- ServiceFabric test clusters
  - 2 test clusters: 3 D2 and 3 D3 machines
  - 3 services (GatewayActor, NodeActor, PointActor), each with 3 partitions and 3 replicas
  - KVS State provider, each invocation persists Actor’s state

Lessons Learned

- Cohesive logging is critical and hard
  - Need big picture for distributed services
  - Semantic filtering is better than timeline
- Take time to pick VM class
- CPU/Memory 300k sensor actors
  - Fan out
- DB “Broker” is huge help
  - Minimize trips to the database

Lessons Learned (continued)

- Density/Cost - At small scale can have higher costs. At larger scale Service Fabric more cost effective
  - For MeshVista, 5 A3 machines cost more (approx.) 10-20% than existing cloud services solution.
- Be careful with actor programming model: Simplicity over performance
  - Careful of actor granularity. Don’t go too granular or too coarse
  - Lots of message passing between actors will affect performance
  - Care of single threaded concurrency – calling “child” actor, blocks calls on “parent” actor. This can lead to longer fan-out times
  - Querying actors should be limited. Actors are great for isolated “avatars” but not good where data has to be aggregated or to build a view

Microsoft

Service Fabric - Use Case Scenarios and Examples

Illyriad Games

Microsoft Services

ILLYRIAD® GAMES

Independent games developer specializing in multiplayer games

Micro-studio with 5 core members of staff

Age of Ascent

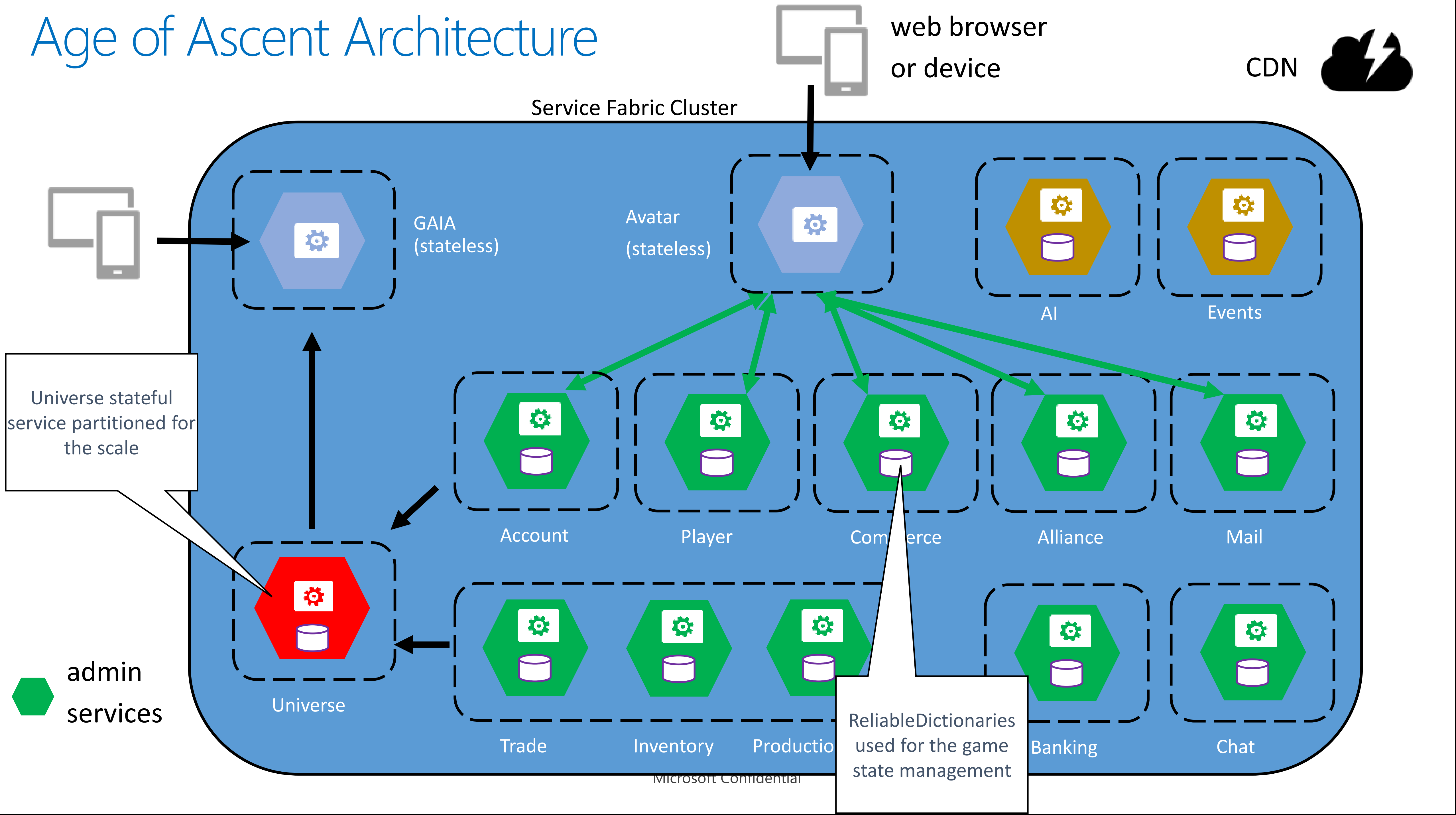
- Born in the cloud – A new scale of gaming
- Space twitch combat, empire building, player driven economy
- Millions of players concurrently in single contiguous universe
- 50,000+ simultaneous players, real-time in the same battle
- 267 million network messages per second (23 Trillion per day)

“Age of Ascent is poised to become something special, one of those games that players will point to years down the road and say, *‘Yep, that was the one that changed everything‘.*” - MMORPG.com

Age of Ascent Technology

- Use Microsoft Azure for hyperscale
- ASP.NET Core x64
  - Platform Agnostic – HTML5 + JS + WebGL Client, Azure Backend
  - Anywhere, Anytime – Cross device
- Service Fabric – game state management
- Listen to Hanselminutes podcast - Inside Age of Ascent with Ben Adams
  - http://hanselminutes.com/509/inside-age-of-ascent-with-ben-adams

Age of Ascent Architecture



Universe stateful service partitioned for the scale

ReliableDictionaries used for the game state management

Service Fabric Benefits for Age of Ascent

- Deployment and upgrade agility
- Cost and efficient utilization due to high density
    - Illyriad estimated they would need ~700 cloud service machines to run Age of Ascent. On Service Fabric can achieve the same with < 100
- Elasticity scale in and out - manage costs
    - Dynamic Scaling – Spatial unfolding and refolding with scale sets
- Cluster management and orchestration
    - Dynamic Load Distribution – Service Fabric balancing of hot and cold nodes
- Always available stateful microservices

Key Service Fabric Features

- Actors = In memory device avatars
- Lifecycle management
  - Rolling app independent updates
- Application specific horizontal scalability
  - Scale apps independent of compute (Density)
- Reliable collections – cache in place
  - Database load/connection throttling
- Lightweight applications by concern

In Review: Takeaways

- Always have a clear and concise understanding of what you mean by scale and agility for your particular scenarios
- Service Fabric is a platform for building applications with a microservices approach
- See other case studies at http://blogs.msdn.com/b/azureservicefabric/

Lab: Using Service Fabric to build a customer order application