

# Growing the Business on the Mainframe.



**Misha Kravchenko - Marriott  
VP Mainframe Delivery**

**David Morley - Marriott  
z/TPF Technical Architect**

**TPF User Group – Denver 2019  
Keynote Session**

# AGENDA

## Marriott

- Starwood Merger
- z/Story at Marriott – High Level Architecture Overview

## Why MongoDB?

- Shopping and other demands on our system
- Create a Team – Partners with products and services

## Our POC

- And a look into production and beyond  
NOTE: Statements of possible direction do not equal any commitments

Q&A encouraged throughout the session.

# AGENDA

## Marriott

- **Starwood Merger**
- z/Story at Marriott – High Level Architecture Overview

## Why MongoDB?

- Shopping and other demands on our system
- Create a Team – Partners with products and services

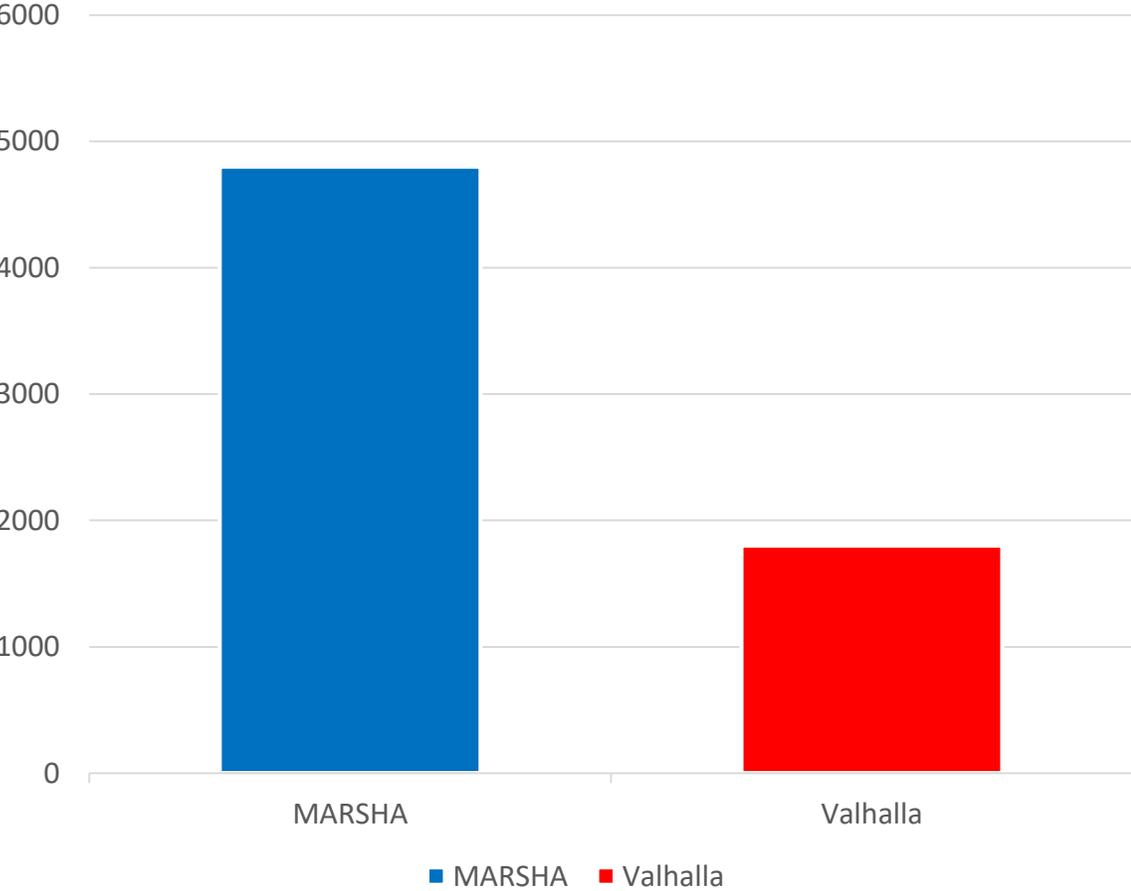
## Our POC

- And a look into production and beyond
- NOTE: Statements of possible direction do not equal any commitments

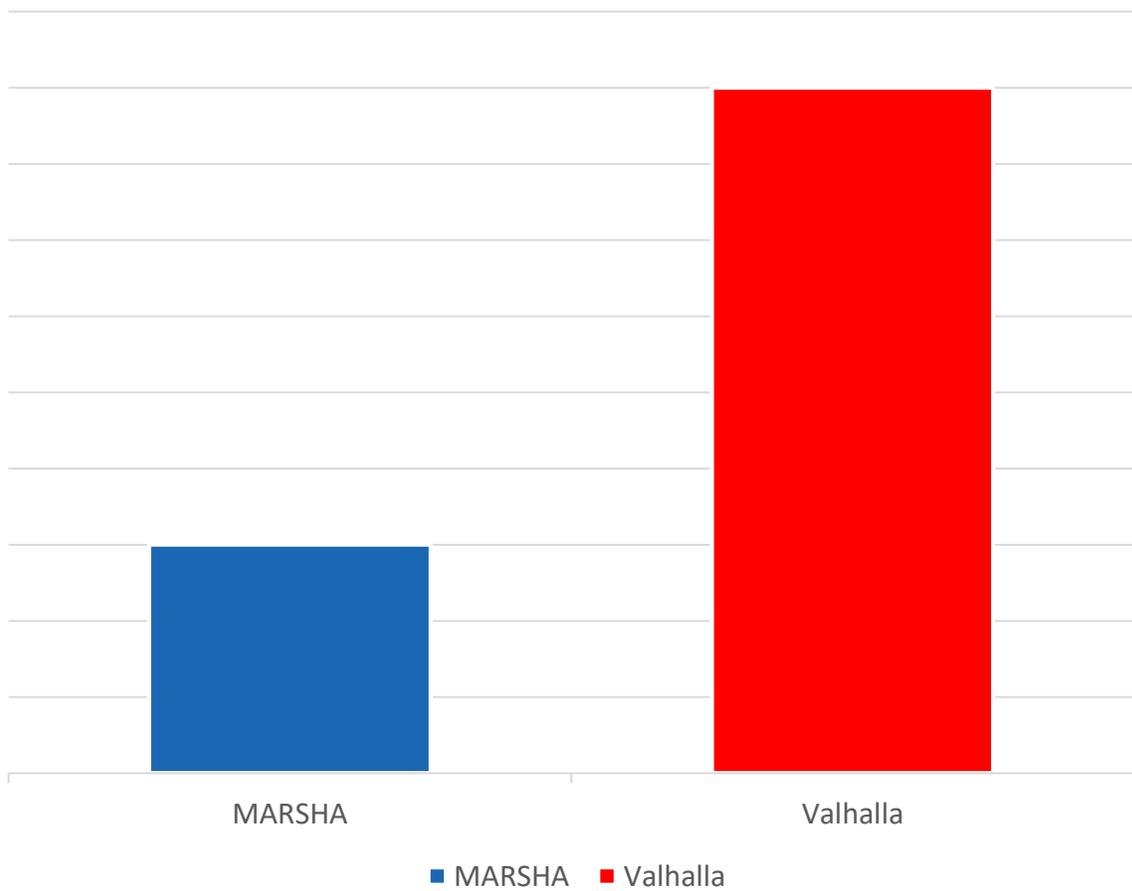
**Q&A encouraged throughout the session.**

# Pre-Merger MARSHA and Valhalla in 2016

Properties

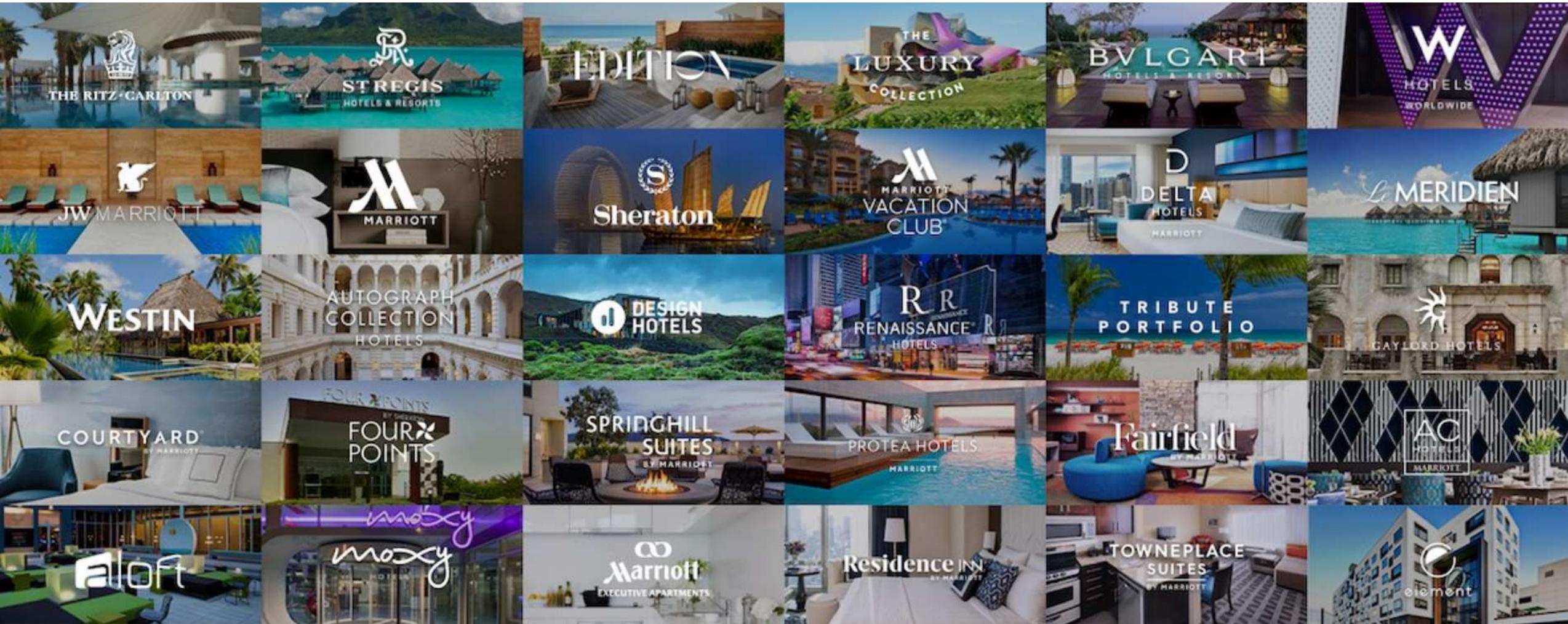


Costs



# Over 6,900 Properties in 130 Countries across 30 brands

Still growing by approximately two properties a day



# AGENDA

## Marriott

- Starwood Merger
- **z/Story at Marriott – High Level Architecture Overview**

## Why MongoDB?

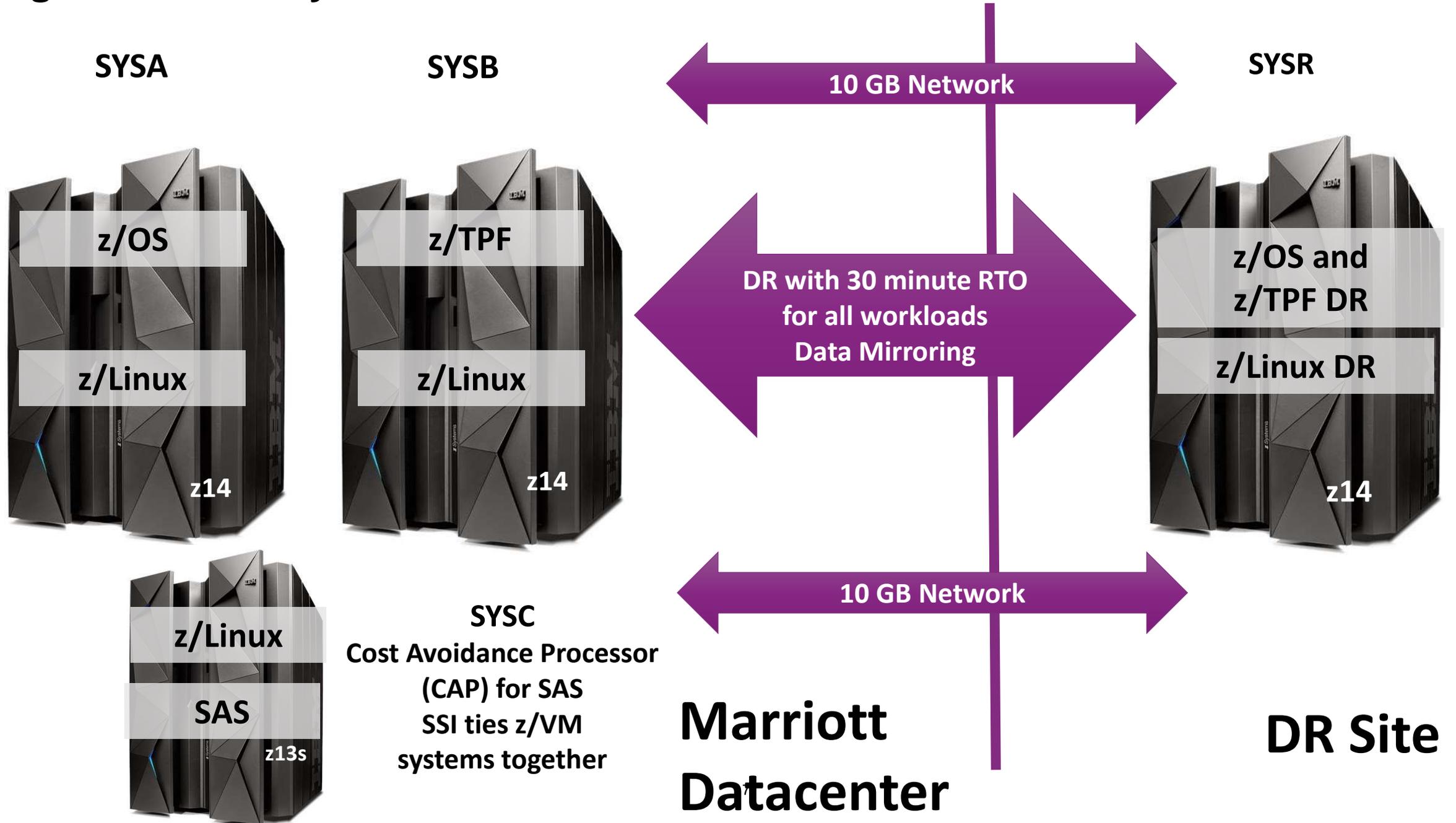
- Shopping and other demands on our system
- Create a Team – Partners with products and services

## Our POC

- And a look into production and beyond
- NOTE: Statements of possible direction do not equal any commitments

**Q&A encouraged throughout the session.**

# High Level z/Story at Marriott



# AGENDA

## Marriott

- Starwood Merger
- z/Story at Marriott – High Level Architecture Overview

## Why MongoDB?

- **Shopping and other demands on our system**
- Create a Team – Partners with products and services

## Our POC

- And a look into production and beyond
- NOTE: Statements of possible direction do not equal any commitments

**Q&A encouraged throughout the session.**

# Shopping Traffic, Bad Calls, Dynamic Pricing... and the impacts to MARSHA, our reservation system



- Rate information is collected and cached by whole sale distributors for public consumption.
- All travel and transportation services need to have this type of data available today.



Best Prices Guaranteed at [Marriott.com](https://www.marriott.com)



# Revenue Management (RM) and Shopping Activity

Automated Revenue Management facilitates repricing of inventory in near real-time.

Update frequency has gone from once every 3 days to 3 times a day.

For every 4% increase in RM activity we get a ~2.5% increase in shopping activity, mostly from 3<sup>rd</sup> parties.

RM activity has increased 400% since January 2017.

# CAIRO?

Cached Availability Inventory and Rate Offers



# 60/90 Day-cycles

Focus on delivering in smaller chunks.

Summer of Performance (90 days)

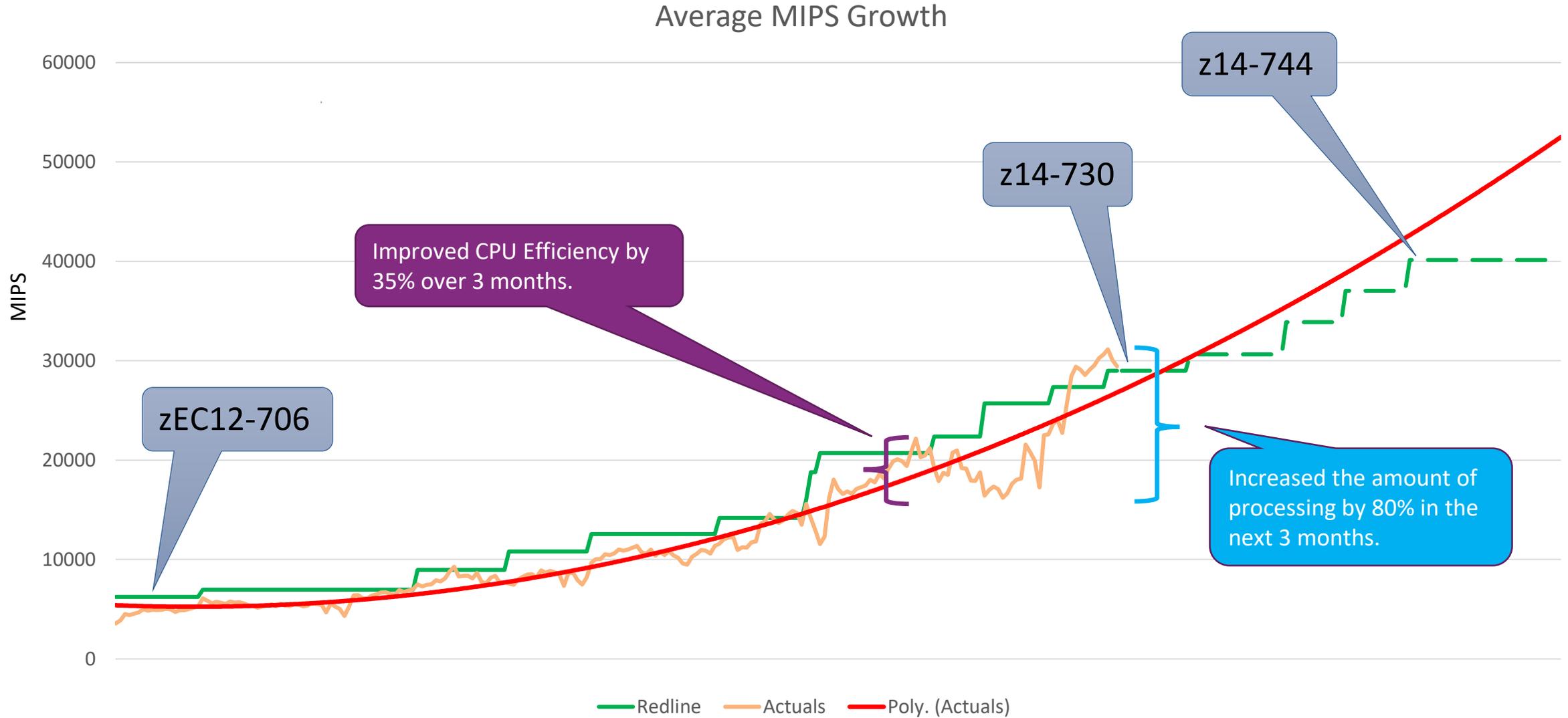
CAIRO Proof of Concept (60 days)

Migrate Starwood onto MARSHA (90 days)

CAIRO Deliver Production (90 days)

Keep the team engaged, informed and looking forward.

# Average z/TPF CPU + projections 2015-2020



# Not Just CPU

**System Scaling is more than CPU. Since January 2017:**

**I/O Rates from 2M per second to 5.5M per second.**

**Memory Footprint from 20GB to 180 GB.**

**Transaction Costs from 3ms to 5ms**

**Transaction Lifetime from 10ms to 100ms**

**The expansion is organic, from mergers and just because...**

# Solving for 80/80/80 Use Case

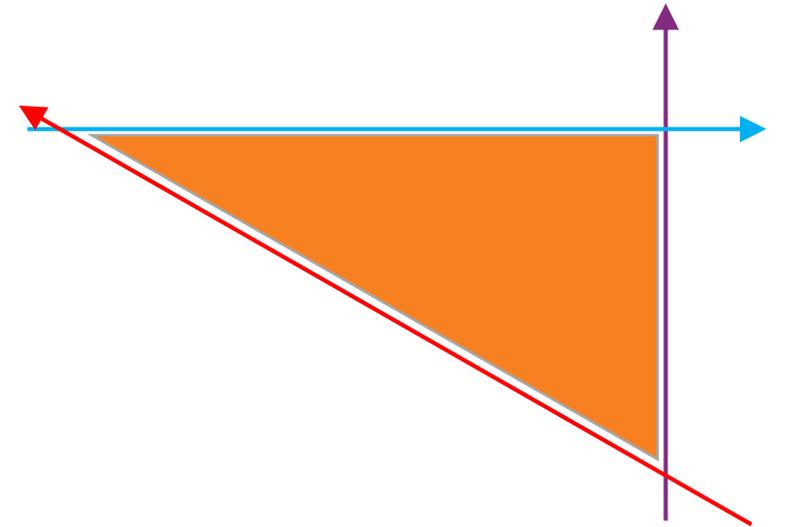
In the universe of possible queries, there is an identifiable subset.

80/20 split are non-member vs member.

80/20 split are for single room vs multi-room.

80/20 split are short stay (<7 nights) vs long stay.

This subset is a candidate for offload without a “boil the ocean” solution.



# AGENDA

## Marriott

- Starwood Merger
- z/Story at Marriott – High Level Architecture Overview

## Why MongoDB?

- Shopping and other demands on our system
- **Create a Team – Partners with products and services**

## Our POC

- And a look into production and beyond
- NOTE: Statements of possible direction do not equal any commitments

**Q&A encouraged throughout the session.**

# The Team

## Marriott

- Misha Kravchenko – Executive Sponsor
- Mainframe run z/Linux and z/TPF teams

## IBM

- IBM helped simplify contracting, organize resources and driving the overall project plan

## MongoDB

- The MongoDB on z initiative made it a natural choice for this workload.
- MongoDB Team brought invaluable skills.

## Velocity Software

- This team went above and beyond getting MongoDB metrics integrated into zVIEW
- zPRO quickly turns z Systems into the cloud platform mainframe virtualization really created

## Sine Nomine Associates

- Extra support for z/VM and Linux on z Systems
- Coding, including education using Node.js, integrating along with packaging (RPM), testing and 24x7 support for the environment. A bridge between all tech and vendors.

# AGENDA

## Marriott

- Starwood Merger
- z/Story at Marriott – High Level Architecture Overview

## Why MongoDB?

- Shopping and other demands on our system
- Create a Team – Partners with products and services

## Our POC

- **And a look into production and beyond**  
**NOTE: Statements of possible direction do not equal any commitments**

**Q&A encouraged throughout the session.**

# A Universal Software Stack

Not everyone in the team is a mainframe developer.  
Nor do they have to be.

GITHub repository + other open source tooling.

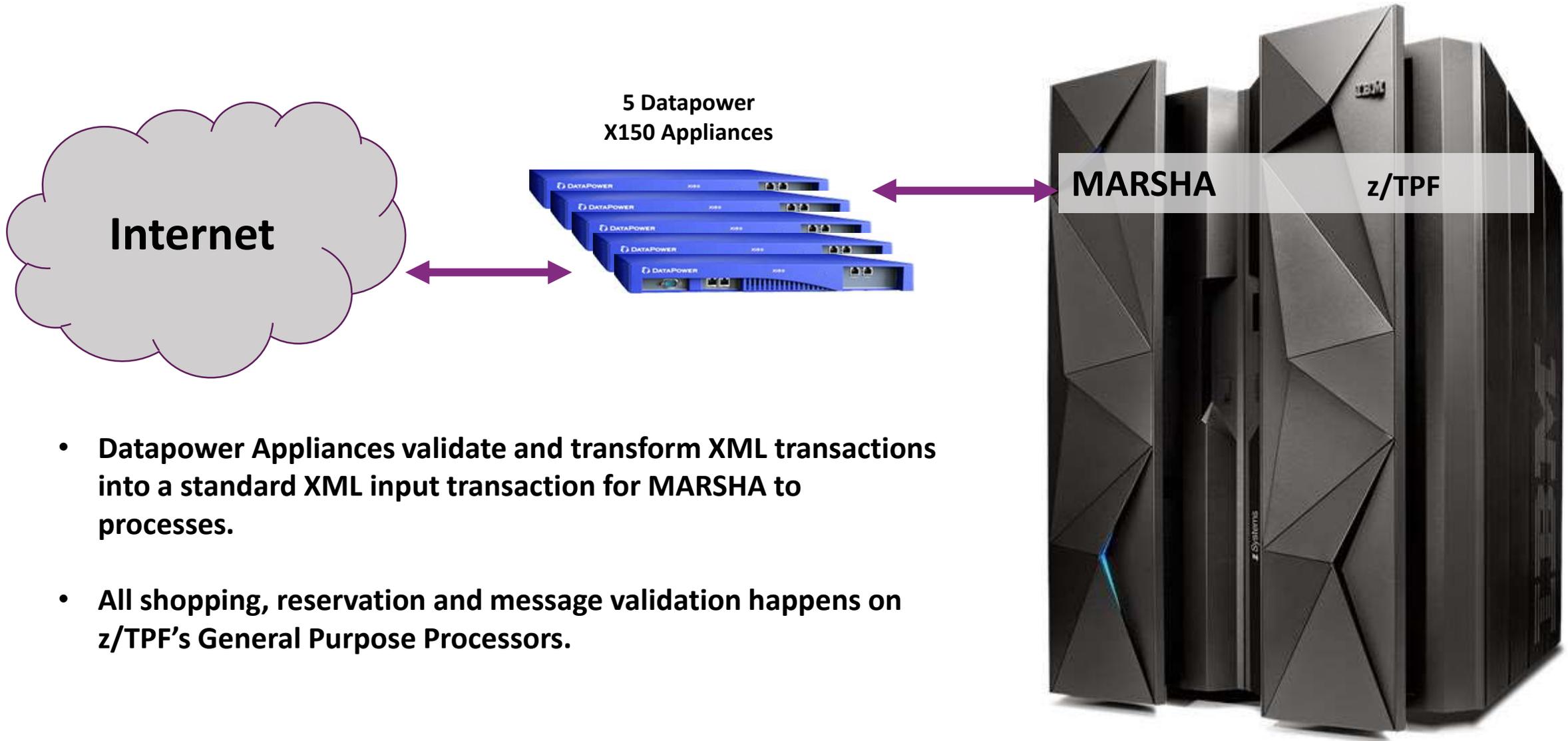
Linux OS

MongoDB Database

Node.js programming language

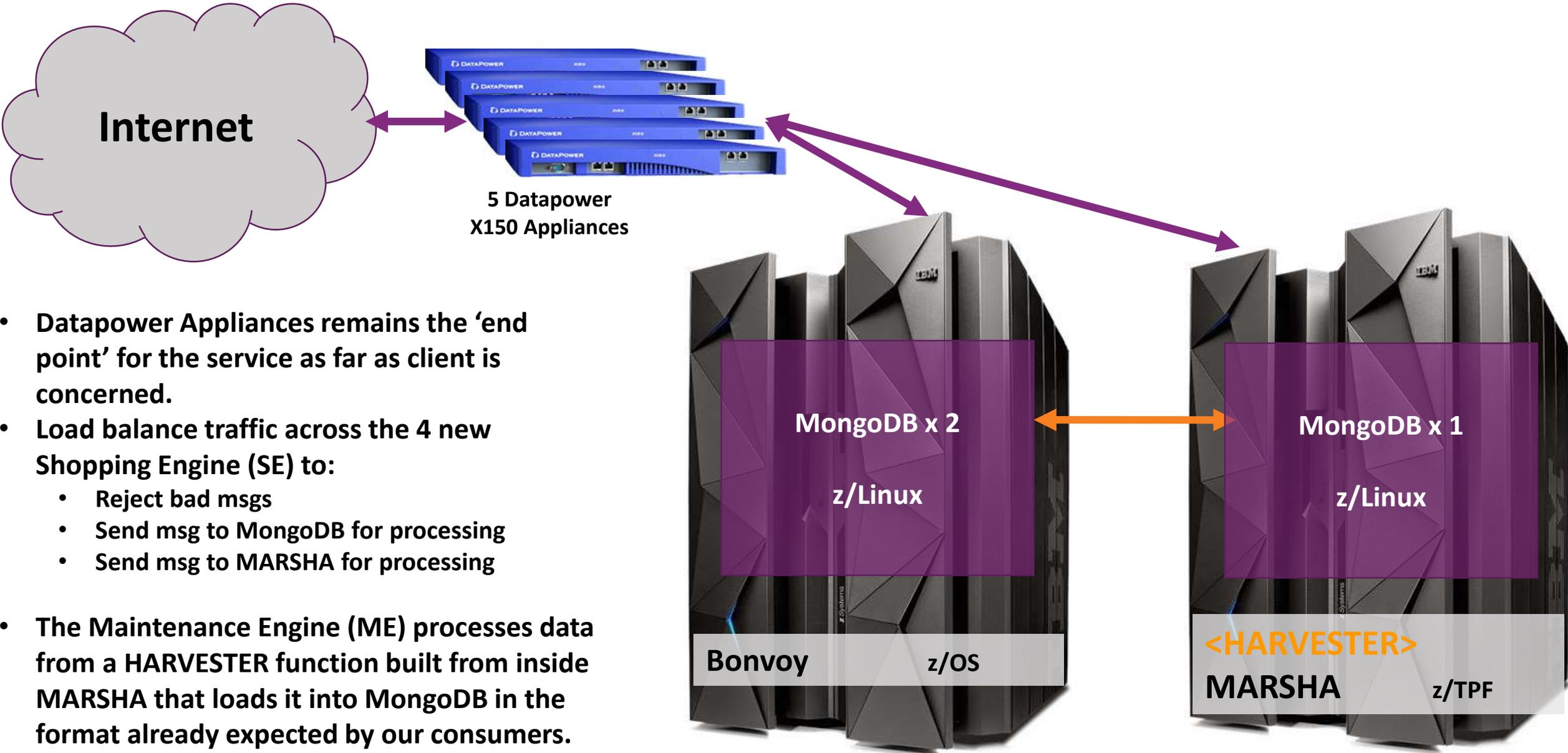
Runs on anything from a toaster to a mainframe, #Serverless ready.

# How Shopping Transactions Are Processed Today



- **Datapower Appliances validate and transform XML transactions into a standard XML input transaction for MARSHA to processes.**
- **All shopping, reservation and message validation happens on z/TPF's General Purpose Processors.**

# How Shopping Transactions Will Be Processed



# The Harvester

**When a transaction is received it is added to a list of 'high volume queries'**

**Original Transaction is responded to as usual.**

**List is processed asymmetrically to populate MongoDB.**

**Processing the list uses an adaptive time-initiated algorithm based on the rate of change.**

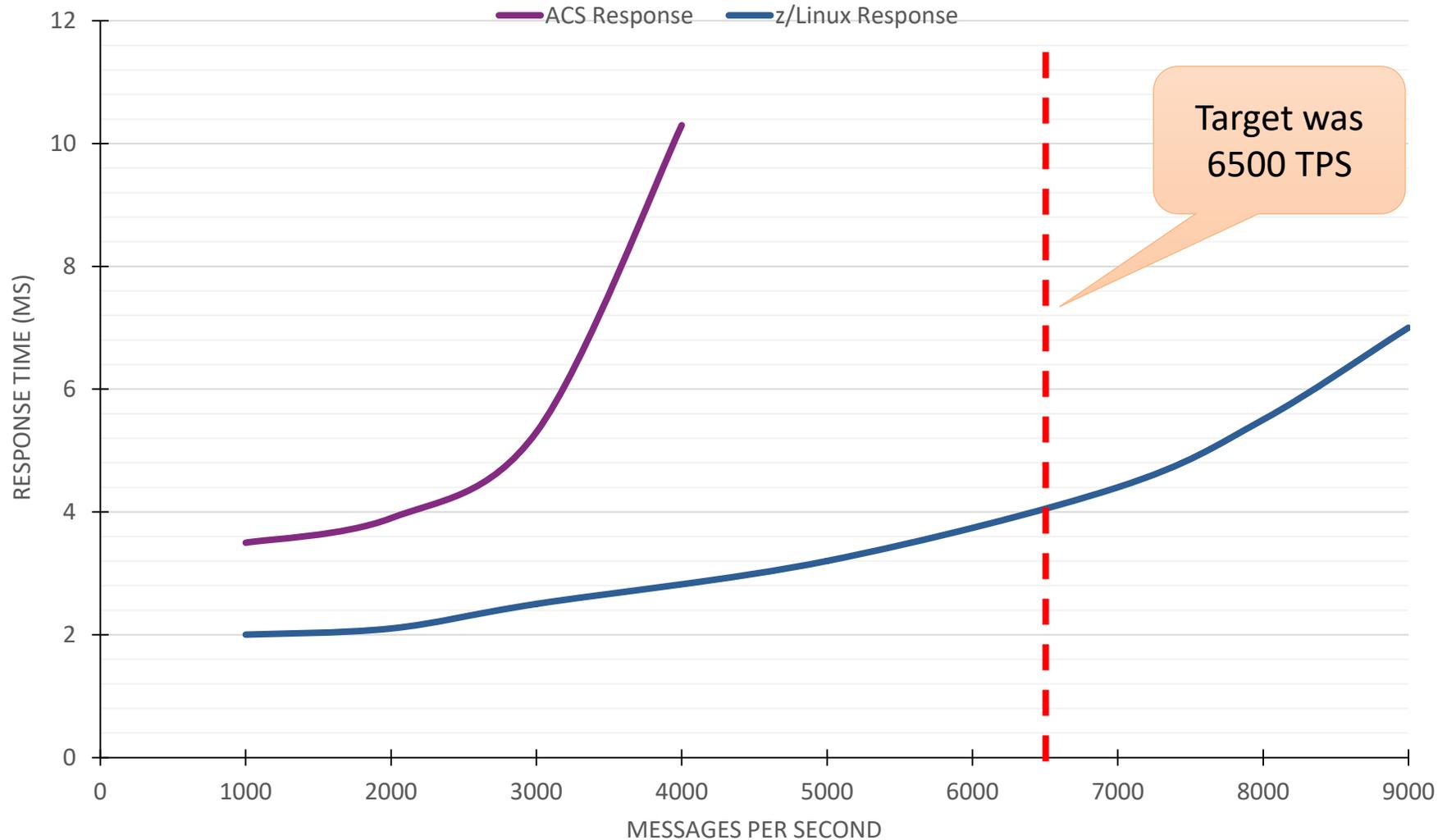
**Target 95% accuracy.**

**The application code is fully reused to ensure consistent results.**

**Shared results across multiple partners will produce 80% of the savings.**

**Efficiencies in the message handling will produce the remainder.**

# 4 Core PoC (not a performance comparison) A Cloud Service (ACS) x86 vs. Linux on z



**1 MongoDB instance on z can handle our required load creating easy failover for HA**

**z/Linux was ~40% faster and speed is key**

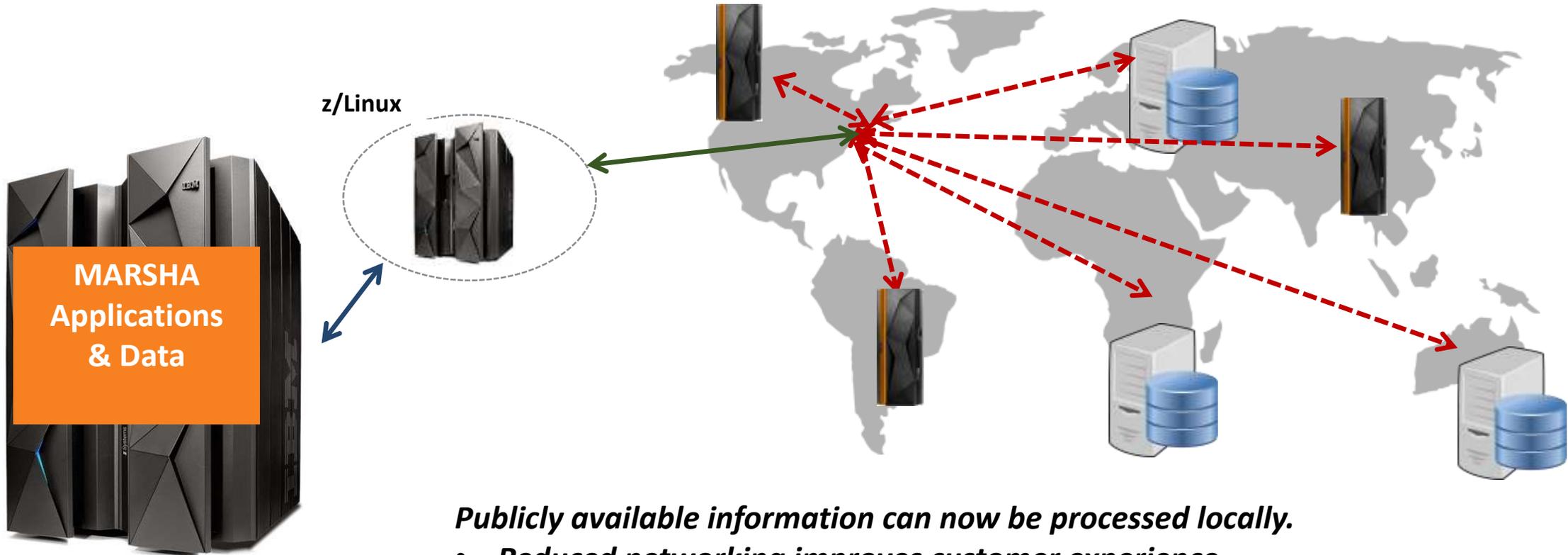
**z/Linux provided ~66% more TPS, bound by 1Gb OSA card not CPU**

**z/Linux degraded gracefully (no errors)**

**ACS streamed errors at 3000 TPS.**

# Possible Future for Federated Shopping

Our local Linux on z MongoDB cluster can be replicated into clouds worldwide.  
The 'nearest' clusters can then automatically be made available to clients around the world.



*Publicly available information can now be processed locally.*

- *Reduced networking improves customer experience*

*Reservations and Rewards processing are still secured on IBM z Systems*

# Conclusions...so far.

A platform agnostic solution that is very z/TPF like in design principles

- NoSQL (document store) – application manages locking

- Asynchronous I/O

- Single Threaded Programming Model

- JIT Compiled Language – only manage source code.

We developed initially in ACS but ran without modification on z/Linux.

System scaled easily and efficiently.