Azure PaaS

Introduction

ZURE

Sakari Nahi

- Coding Executive Officer @ Zure
- Azure MVP
- Twitter @sakarinahi
- Ikkunastudio podcast
  - http://www.ikkunastud.io/
- Finland Azure User Group
  - https://www.meetup.com/Finland-Azure-User-Group/
- Iglooconf
  - https://www.iglooconf.fi/

100%
Azure since 2011

46 / 49
experts

13,7
experience avg.

4,6 / 5
customer satisfaction

3
Azure MVPs

92
employee NPS

Microsoft Partner
Microsoft

Gold Application Development
Gold Cloud Platform
Gold Data Analytics
Gold Data Platform
Gold DevOps

MVP
Microsoft® Most Valuable Professional

Microsoft CERTIFIED Trainer

Partner Seller
Microsoft

Microsoft Partner
2019 Partner of the Year Finalist
Application Innovation Award

ZURE

You can build it with PaaS.

(Reasons against -
expertise, legal/regulation, political, strategy, platform lock-in, time critical)


AI + Machine Learning
Bot Service
Cognitive Search
Cognitive Services
Machine Learning
Microsoft Genomics
Open Datasets
Analytics
Analysis Services
Data Catalog
Data Explorer
Data Lake Analytics
Databricks
Event Hubs
HDInsight
Power BI Embedded
Stream Analytics
Synapse Analytics
Compute
App Service
App Service (Linux)
Azure Functions
Azure VMware Solutions
Batch
Cloud Services
Container Instances
Container Registry
CycleCloud
Dedicated Host
Kubernetes Service
Service Fabric
Databases
Cosmos DB
Data Factory
Database for MariaDB
Database for MySQL
Database for PostgreSQL
Redis Cache
SQL Database
SQL Server Stretch DB
Development
Azure DevOps
DevTest Labs
Lab Services
SignalR Service
Visual Studio App Center
Identity + Security
Azure Active Directory
Azure AD B2C
Azure AD DS
Azure Key Vault
Azure Sentinel
DDoS Protection
Dedicated HSM
Information Protection
Security Center
IoT
Azure Maps
Azure Sphere
Digital Twins
IoT Central
IoT Edge
IoT Hub
Notification Hubs
Time Series Insights
Integration
API Management
Event Grid
Logic Apps
Service Bus
Management + Governance
Automation
Azure Advisor
Azure Arc
Azure Backup
Azure Blueprints
Azure Lighthouse
Azure Monitor
Azure Policy
Azure Portal
Cloud Shell
Cost Management
Managed Apps
Media
Azure CDN
Media Services
Migration
Azure Migrate
Data Box
DB Migration Service
Site Recovery
Networking
Application Gateway
Azure Bastion
Azure DNS
Azure Firewall
Azure Front Door
ExpressRoute
Load Balancer
Network Watcher
Private Link
Traffic Manager
Virtual Network
Virtual WAN
Storage
Avere vFXT
Azure NetApp Files
Azure Storage
Data Lake Storage
Data Share
Managed Disks
StorSimple

App Service

Cosmos DB

Data Lake Storage

Service Bus

Azure Functions

SQL Database

Azure Storage

Event Grid

Azure DevOps

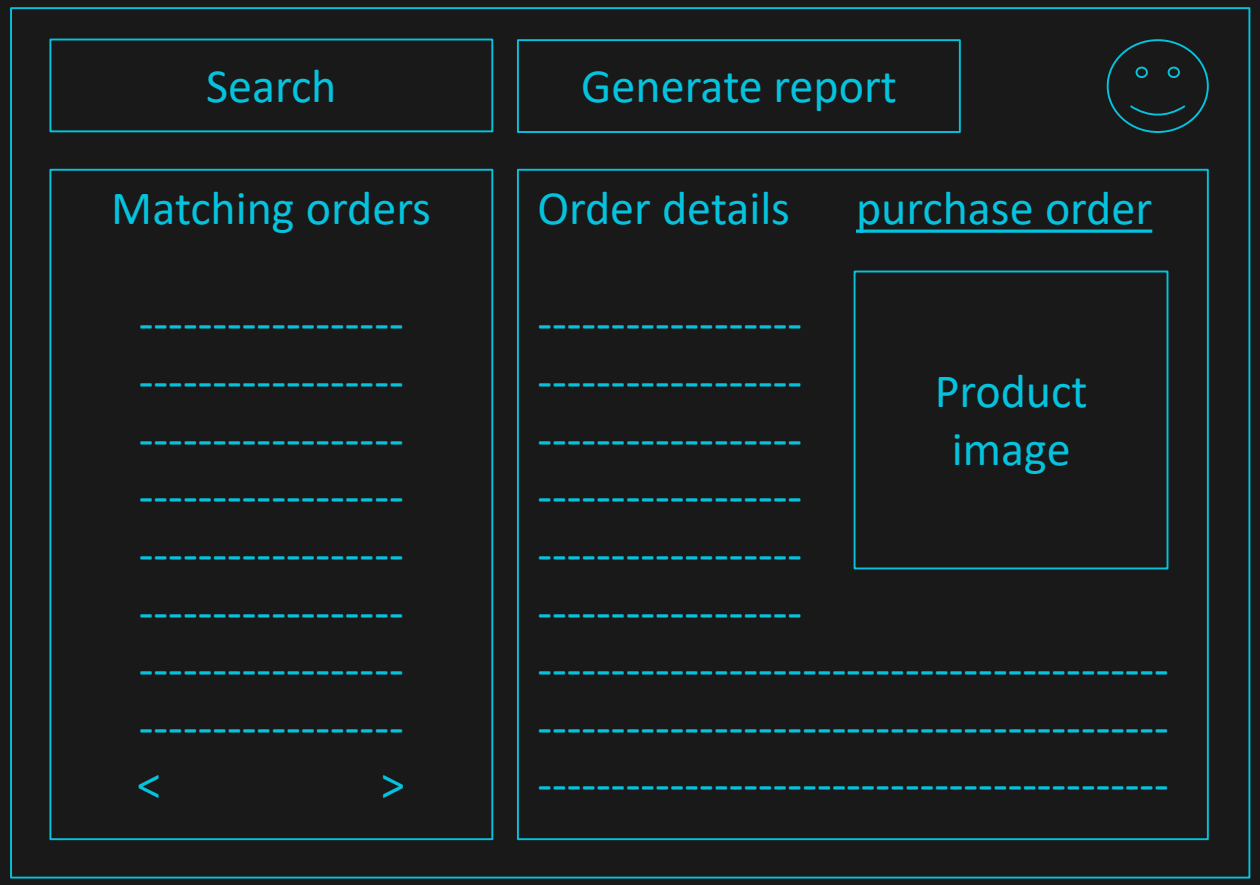
Azure Key Vault

Azure Active Directory

Content

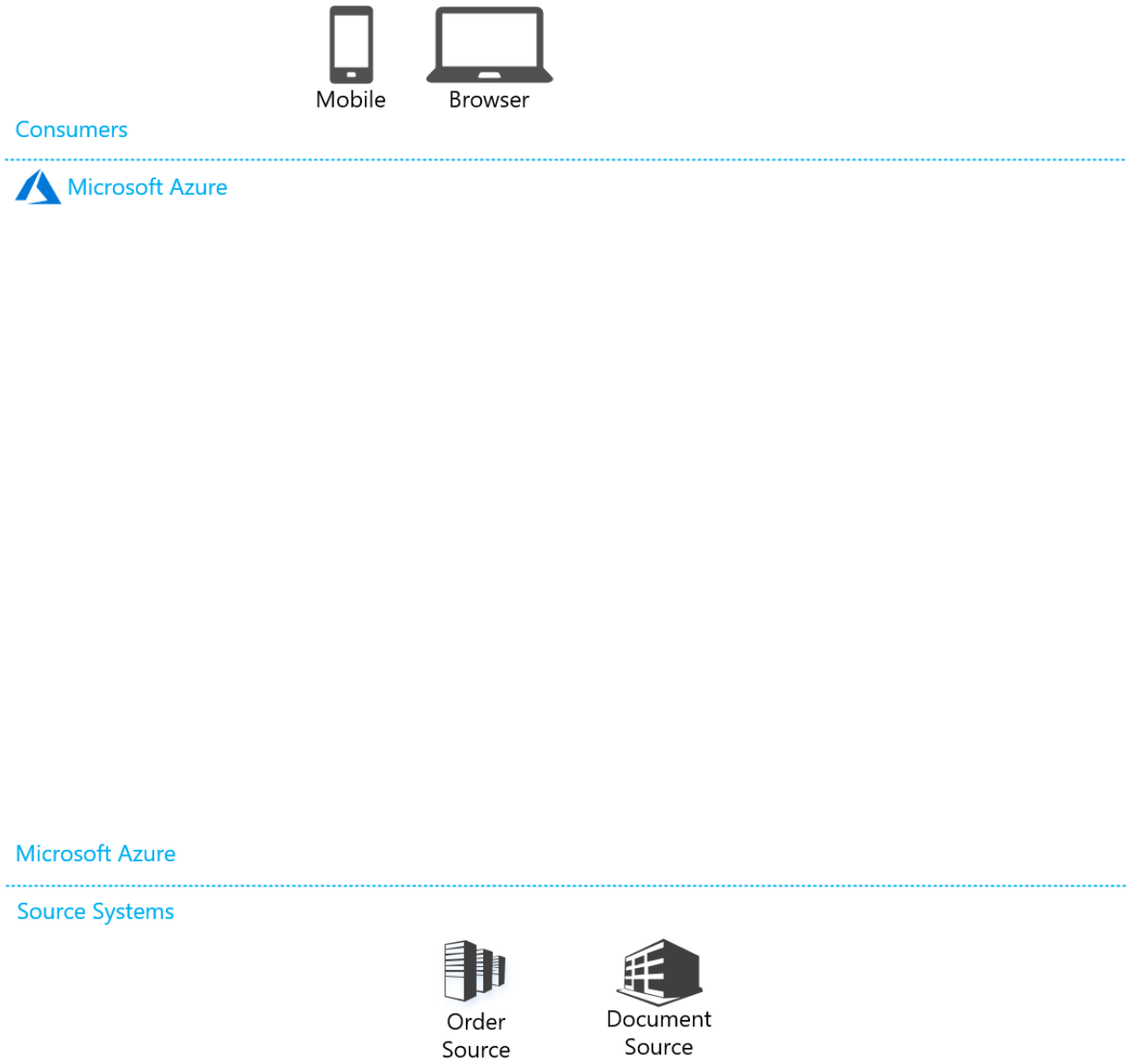
1. Concept
2. Decide
   - Storage
   - Compute
   - Messaging
   - Identity
   - Monitoring

‘Order Master’ Concept



Features

- Users: sales and clients
- Users search for purchase orders in free text
- There are millions of meaningfully diverse orders
- Orders also
  - Refer to sales data: accounts, contacts, activities, etc.
  - Contain binary objects: PO documents and images
- Report generation is compute intensive



Architecture

- We have mobile and browser users
- We have two source systems
  - Order Source contains the orders and the related sales data
  - Document Source contains binaries

Application

Monitoring	Logging, debug data, time series, queryable, custom events, dashboards
Identity	Externals, internals, governance, regulations, SSO, protocols, current IAM
Compute	Control, platform lock-in, infrastructure, scaling, architecture, management
Messaging	Queue, publish/subscribe, routing, persistence, ordering, security, sessions
Storages	Relational, binary, unstructured, caching, analytics, search

Application

Monitoring	Logging, debug data, time series, queryable, custom events, dashboards
Identity	Externals, internals, governance, regulations, SSO, protocols, current IAM
Compute	Control, platform lock-in, infrastructure, scaling, architecture, management
Messaging	Queue, publish/subscribe, routing, persistence, ordering, security, sessions
Storages	Relational, binary, unstructured, caching, analytics, search

Storage

- Single or multiple data store? Multiple.
  - Known as ‘polyglot persistence’
- Is SQL dead? No.
  - ‘SQL doesn’t scale!’ –argument has led into ‘All cloud apps use NoSQL!’
- So... Everything’s on the table, wooo!

Relational ’SQL’	Azure SQL	• Structure & transactions
		• Understood
		  • No horizontal scaling
		  • Dis/assembly costly
		• CRMs, ERPs, etc. relational data

Relational
’SQL’
	Azure SQL	• Structure & transactions
		• Understood
		  • No horizontal scaling
		  • Dis/assembly costly
		• CRMs, ERPs, etc. relational data

Key/value	Cosmos DB Redis Cache Azure Storage	• Hash and value; fast lookups • Scales via duplicates • Caching, sessions, dictionaries, anything single ID

Relational
’SQL’
	Azure SQL	• Structure & transactions
		• Understood
		  • No horizontal scaling
		  • Dis/assembly costly
		• CRMs, ERPs, etc. relational data

Key/value	Cosmos DB Redis Cache Azure Storage	• Hash and value; fast lookups • Scales via duplicates • Caching, sessions, dictionaries, anything single ID

Document
’NoSQL’
	Cosmos DB	• ’Document’ = a collection of fields
		  • Stronger on queries than key/value, scales better than SQL
		  • Documents match application
		• No schema
		• Products, articles, isolated objects

Relational
’SQL’
	Azure SQL	• Structure & transactions
		• Understood
		  • No horizontal scaling
		  • Dis/assembly costly
		• CRMs, ERPs, etc. relational data

Key/value	Cosmos DB Redis Cache Azure Storage	• Hash and value; fast lookups • Scales via duplicates • Caching, sessions, dictionaries, anything single ID

Document
’NoSQL’
	Cosmos DB	• ’Document’ = a collection of fields
		  • Stronger on queries than key/value, scales better than SQL
		  • Documents match application
		• No schema
		• Products, articles, isolated objects

Graph	Cosmos DB	• Edges & Nodes; relationships 1st class • Organization charts, social graphs

Relational
’SQL’
	Azure SQL	• Structure & transactions
		• Understood
		  • No horizontal scaling
		  • Dis/assembly costly
		• CRMs, ERPs, etc. relational data

Key/value	Cosmos DB Redis Cache Azure Storage	• Hash and value; fast lookups • Scales via duplicates • Caching, sessions, dictionaries, anything single ID

Document
’NoSQL’
	Cosmos DB	• ’Document’ = a collection of fields
		  • Stronger on queries than key/value, scales better than SQL
		  • Documents match application
		• No schema
		• Products, articles, isolated objects

Graph	Cosmos DB	• Edges & Nodes; relationships 1st class • Organization charts, social graphs
Search	Azure Search	• Text indexing, aggregates • Catalogs, content search

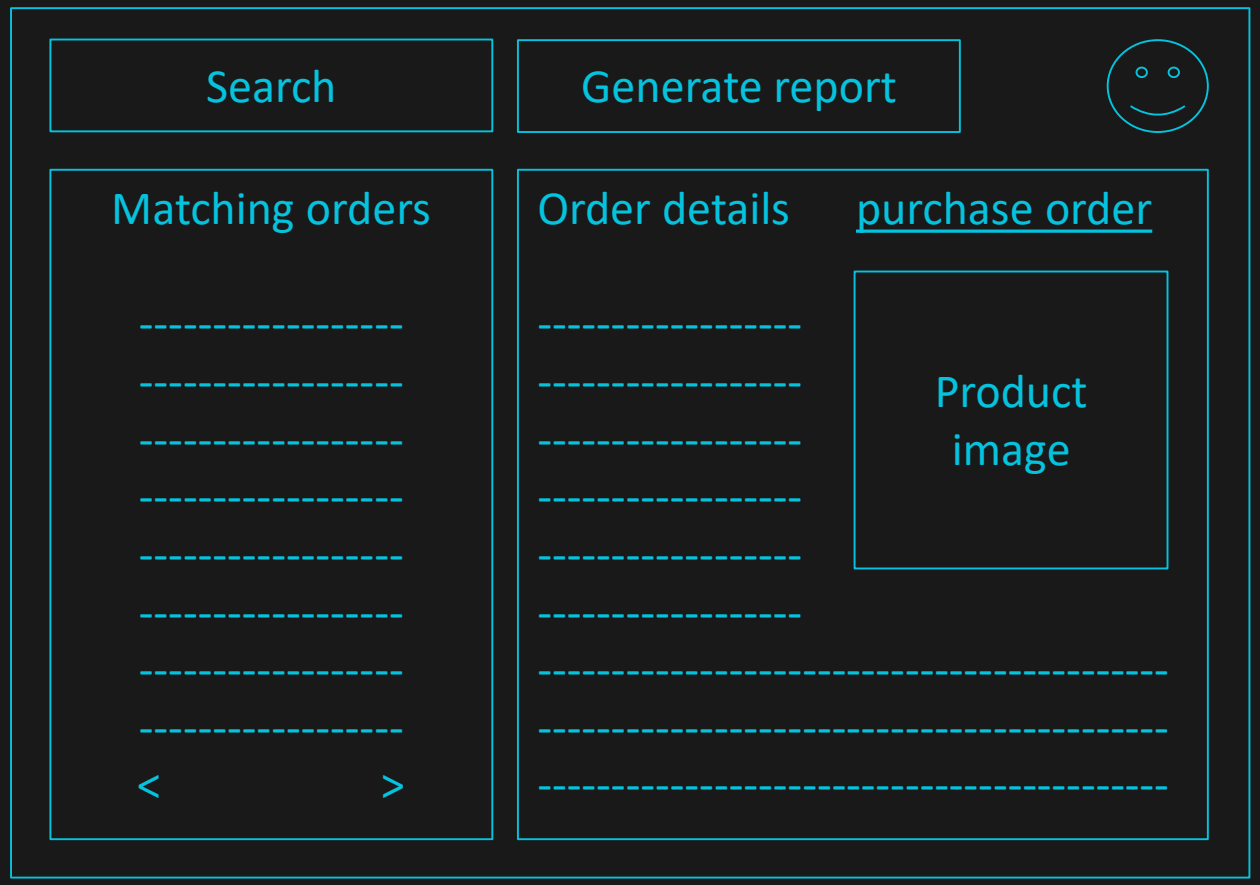
Relational
’SQL’
	Azure SQL	• Structure & transactions
		• Understood
		  • No horizontal scaling
		  • Dis/assembly costly
		• CRMs, ERPs, etc. relational data

Key/value	Cosmos DB Redis Cache Azure Storage	• Hash and value; fast lookups • Scales via duplicates • Caching, sessions, dictionaries, anything single ID

Document
’NoSQL’
	Cosmos DB	• ’Document’ = a collection of fields
		  • Stronger on queries than key/value, scales better than SQL
		  • Documents match application
		• No schema
		• Products, articles, isolated objects

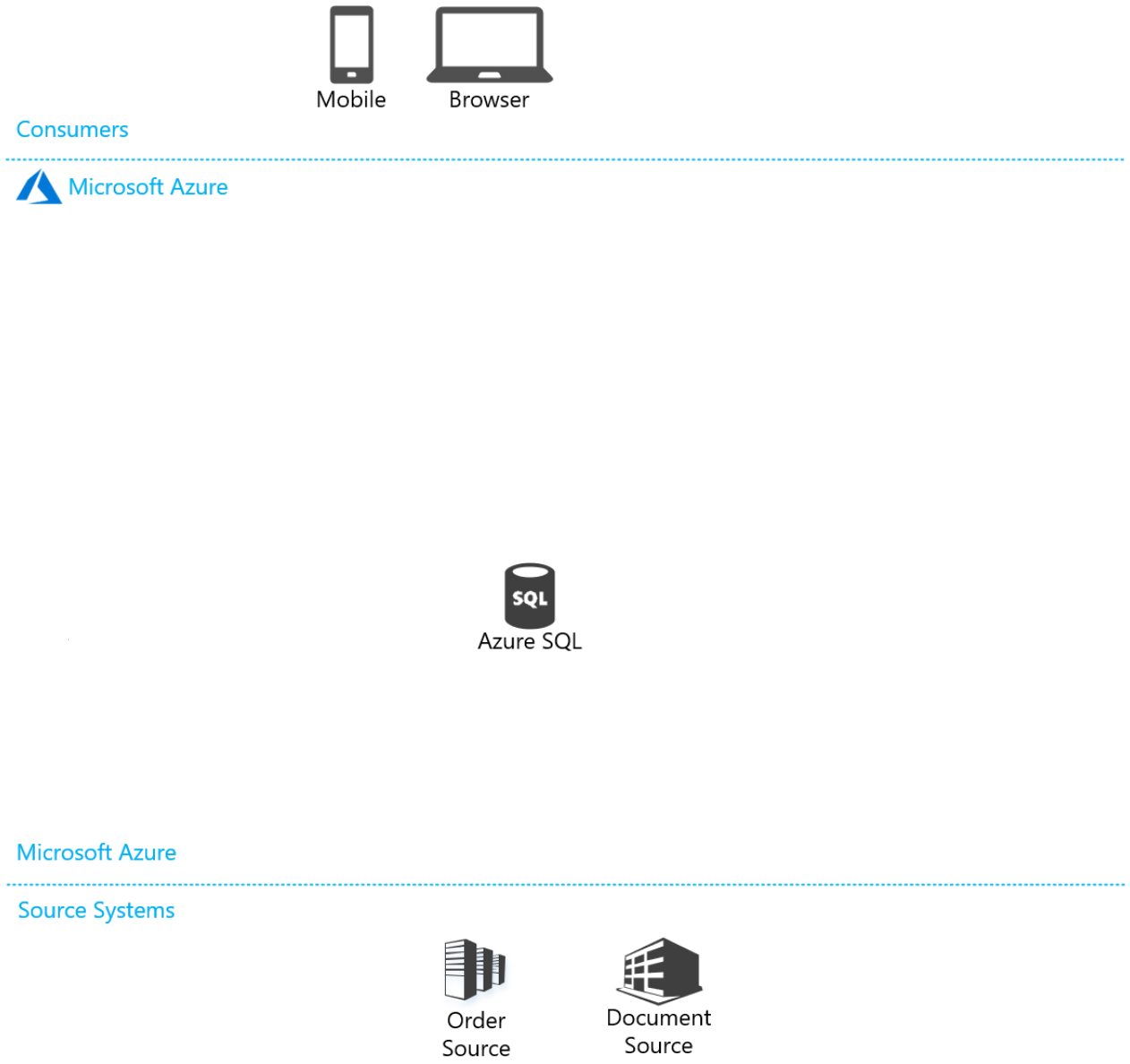
Graph	Cosmos DB	• Edges & Nodes; relationships 1st class • Organization charts, social graphs
Search	Azure Search	• Text indexing, aggregates • Catalogs, content search
Binary	Blob Storage Data Lake Store	• Binaries / structureless • Images, CSV’s, logs, raw JSON-data

‘Order Master’ Concept



Features

- Users: sales and clients
- Users search for purchase orders in free text
- There are millions of meaningfully diverse orders
- Orders also
  - Refer to sales data: accounts, contacts, activities, etc.
  - Contain binary objects: PO documents and images
- Report generation is compute intensive



Architecture

Relational sales data in Azure SQL

Accounts, contacts, leads, opportunities etc. are very relational and it is likely we'll be querying them in complex manner

Dynamic orders in Cosmos DB

The products vary from customer to customer, making it hard to fit orders into a single schema

Binaries and raw source data in Azure Data Lake

Concept's access control requires POSIX and the client is looking forward to doing analytics

Free text capabilities in Azure Search

A plethora of capabilities, cheaper than Cosmos

Caching in Redis

Application

Monitoring	Logging, debug data, time series, queryable, custom events, dashboards
Identity	Externals, internals, governance, regulations, SSO, protocols, current IAM
Compute	Control, platform lock-in, infrastructure, scaling, architecture, management
Messaging	Queue, publish/subscribe, routing, persistence, ordering, security, sessions
Storages	Relational, binary, unstructured, caching, analytics, search

Cloud, now

Self-managed	Self-managed Unit of Scale	Cloud-managed

On-prem	IaaS	CaaS	PaaS	FaaS	SaaS
Functions	Functions	Functions	Functions	Functions	Functions
Application	Application	Application	Application	Application	Application
Runtime	Runtime	Runtime	Runtime	Runtime	Runtime
Containers	Containers	Containers	Containers	Containers	Containers
Operating System	Operating System	Operating System	Operating System	Operating System	Operating System
Virtualization	Virtualization	Virtualization	Virtualization	Virtualization	Virtualization
Hardware	Hardware	Hardware	Hardware	Hardware	Hardware

ZURE

Compute

IaaS

For absolute control, with the most work.

CaaS

Lift old stuff, or ensure freedom of movement between platforms.

PaaS & FaaS

Cloud native approach, with the least administration.

Most control	Least control
Most admin	Least admin
Slowest TTM	Fastest results
Hardest to scale	Easy to scale
Existing knowledge	New knowledge

Compute

IaaS

For absolute control, with the most work.

Virtual Machines
+ other services

(+ networking, managed disks, site recovery, OS administration, etc.)

CaaS

Lift old stuff, or ensure freedom of movement between platforms.

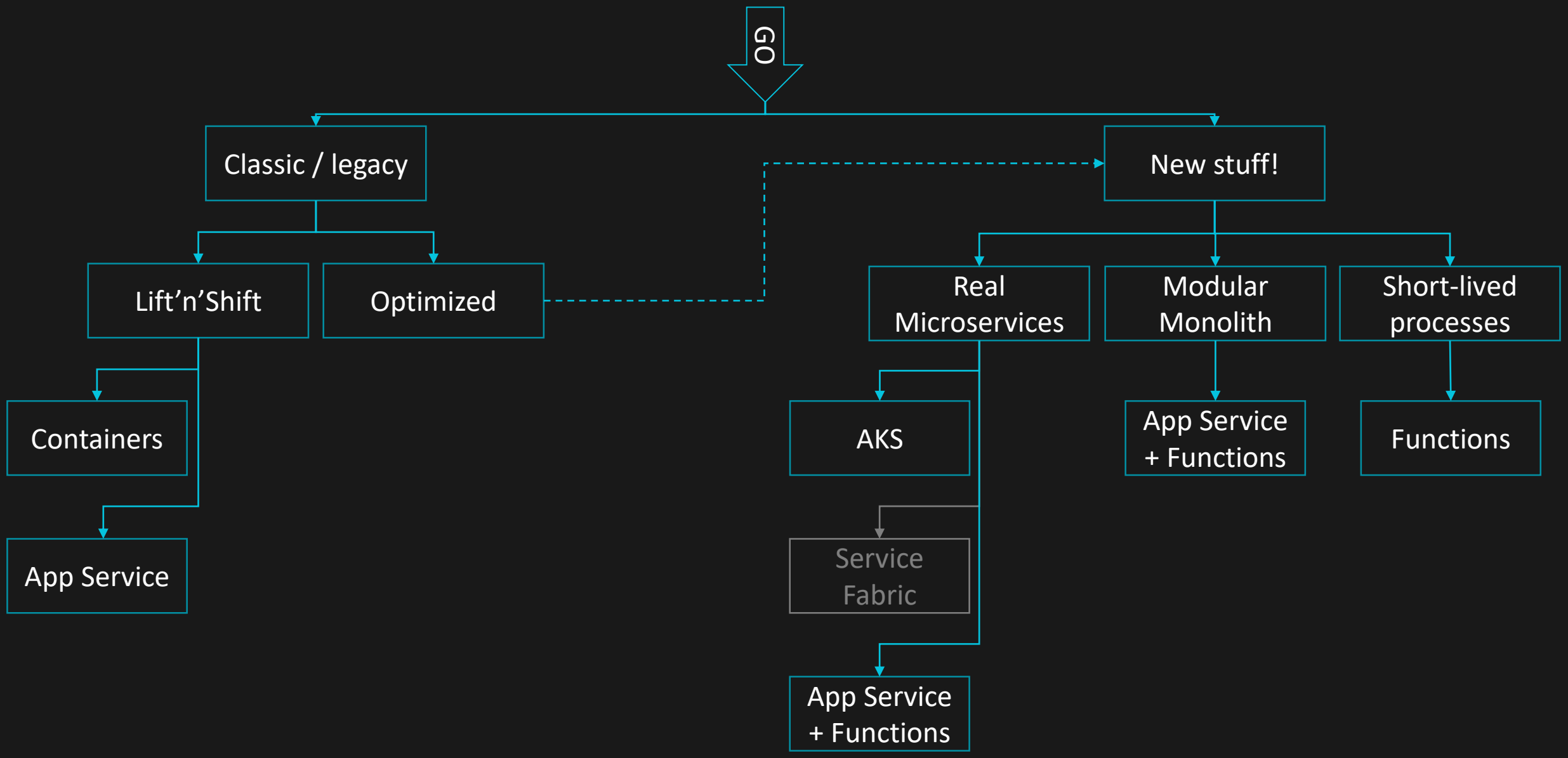
Azure Kubernetes Service
Azure Container Instance
App Service

(+ networking, container registry, load balancers, runtime updates, etc.)

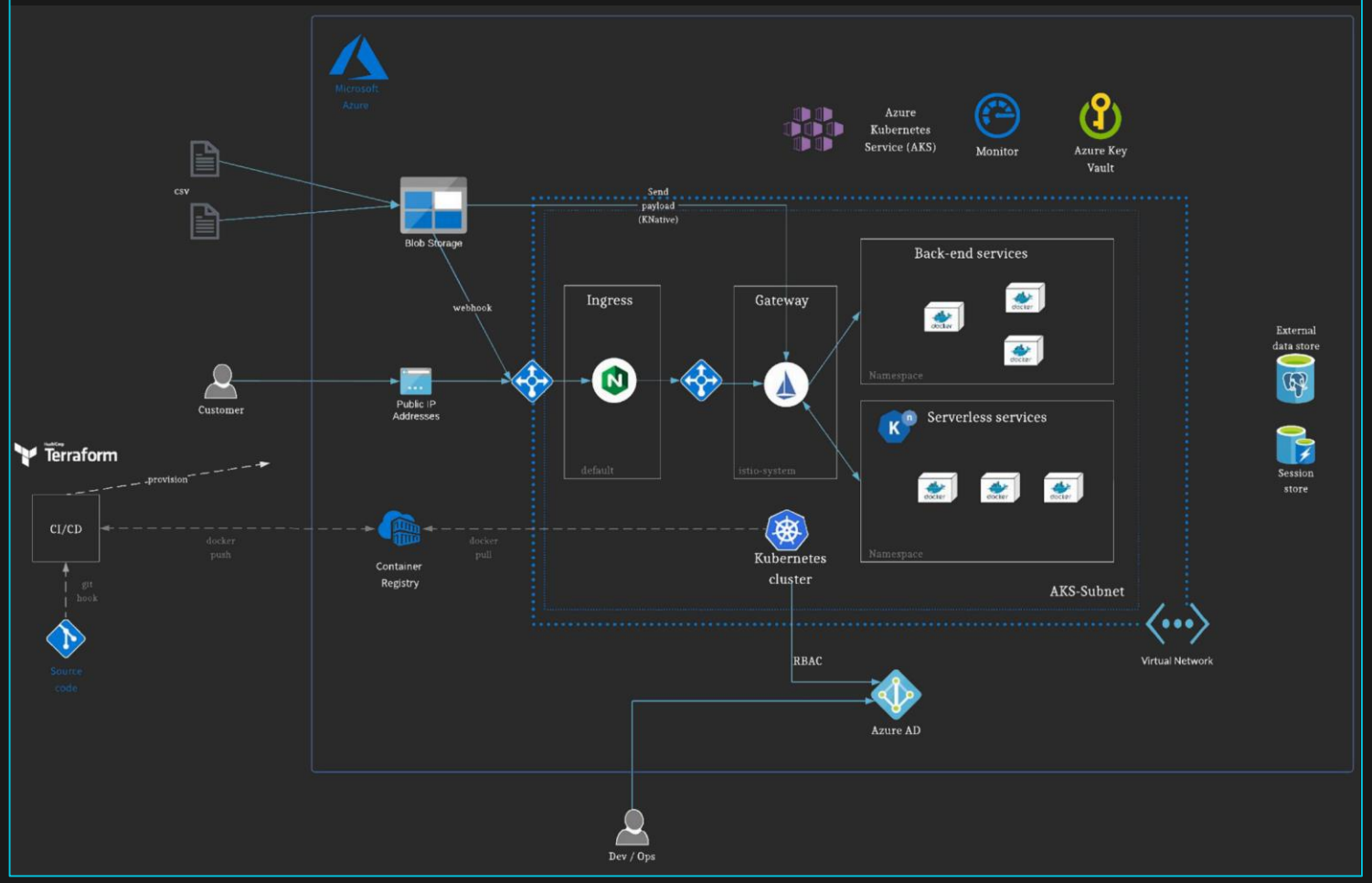
PaaS & FaaS

Cloud native approach, with the least administration.

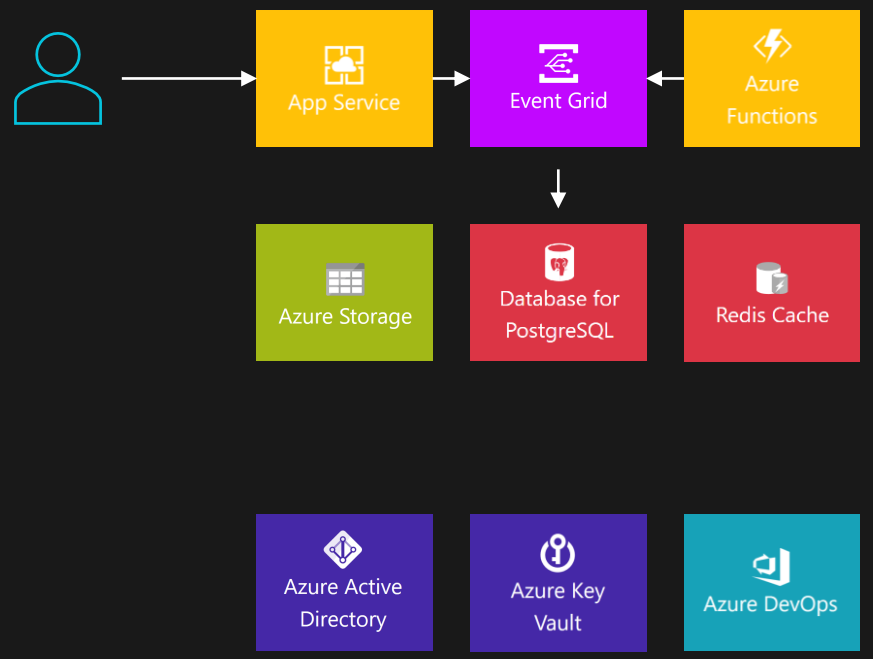
App Services
Azure Functions
Service Fabric


GO
Classic / legacy
New stuff!
Lift'n'Shift
Optimized
Real Microservices
Modular Monolith
Short-lived processes
Containers
App Service
AKS
Service Fabric
App Service + Functions
App Service + Functions
Functions

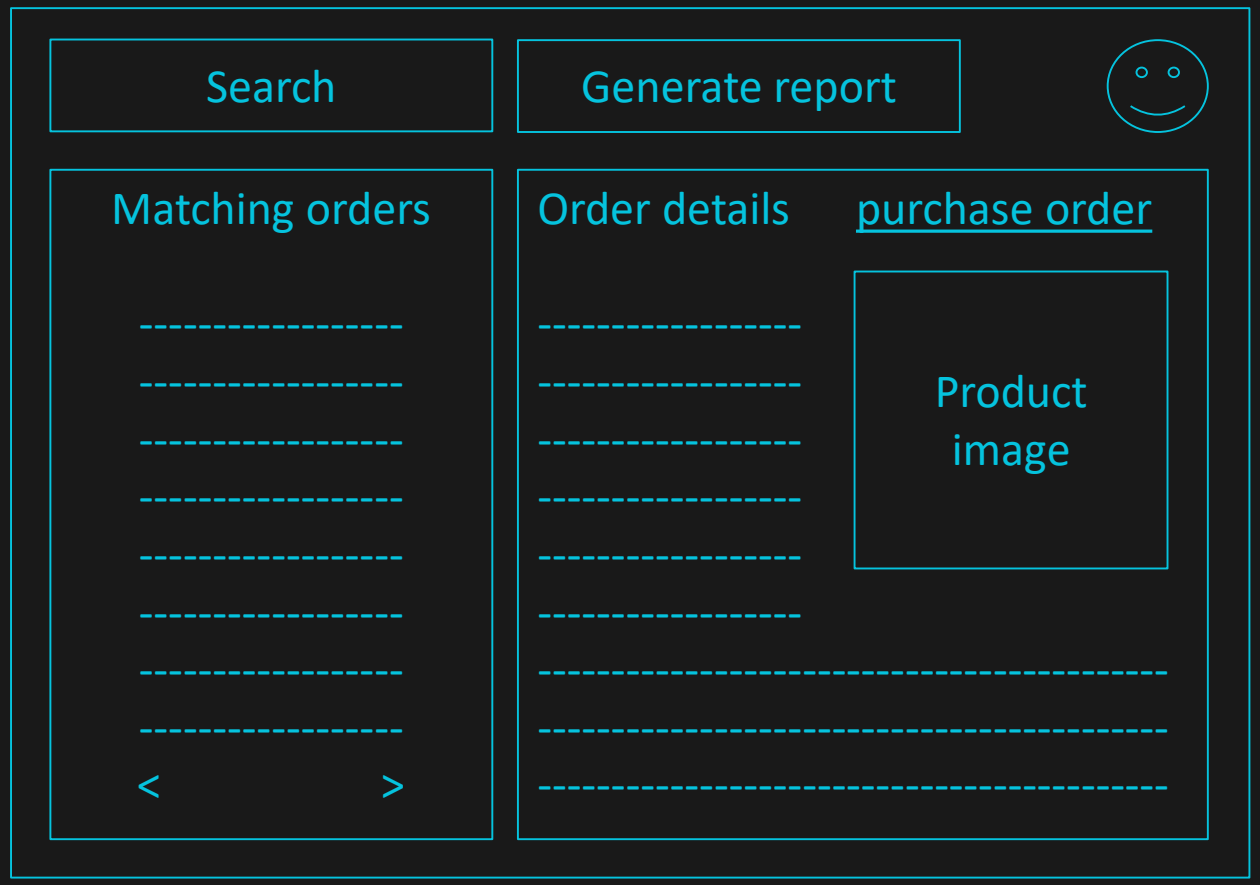
Kubernetes



PaaS

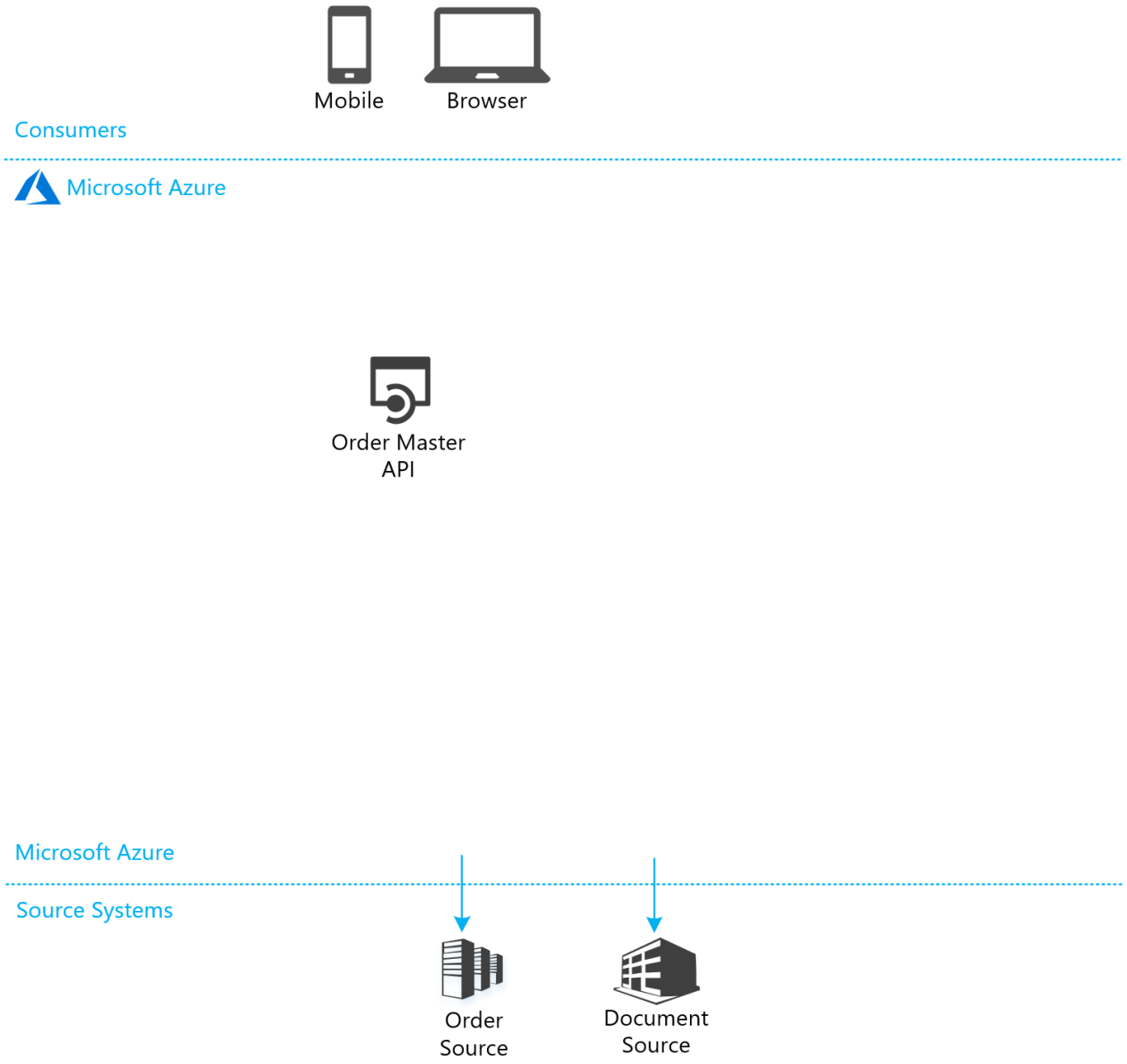


‘Order Master’ Concept



Features

- Users: sales and clients
- Users search for purchase orders in free text
- There are millions of meaningfully diverse orders
- Orders also
  - Refer to sales data: accounts, contacts, activities, etc.
  - Contain binary objects: PO documents and images
- Report generation is compute intensive



Architecture

Web App to respond to requests

We want this guy to be online 100%

Function handles intensive report generation

Let's not eat the resources from the API

Function reads Order Source

Persists orders into Cosmos DB and sales data into Azure SQL as scheduled; probably puts raw data into Data Lake as well

Data Factory for basic integration needs

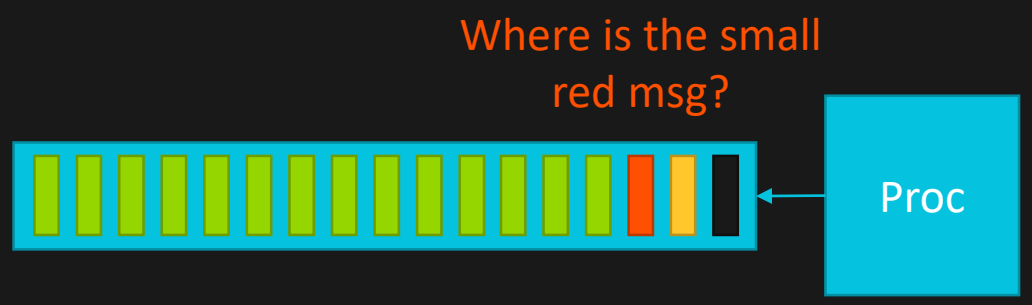
'Copy' from Document Source to Data Lake

Application

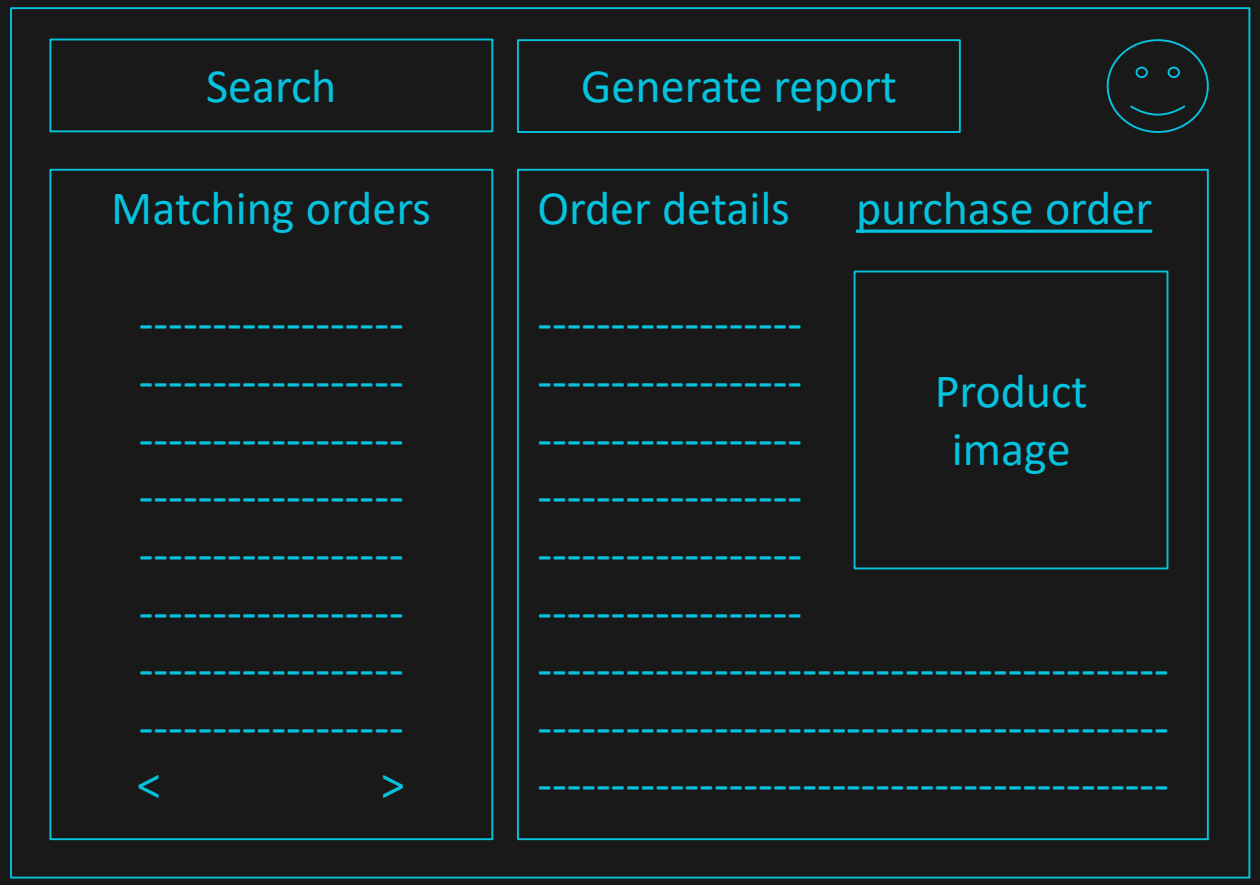
Monitoring	Logging, debug data, time series, queryable, custom events, dashboards
Identity	Externals, internals, governance, regulations, SSO, protocols, current IAM
Compute	Control, platform lock-in, infrastructure, scaling, architecture, management
Messaging	Queue, publish/subscribe, routing, persistence, ordering, security, sessions
Storages	Relational, binary, unstructured, caching, analytics, search

Messaging

- Storage Queue – "I just want to make sure everything is processed!"
  - Persist messages in a queue, let someone pull message and process
  - Large queues (80GB+) okay!
- Azure Event Grid – "Fast, cheap, scary!"
  - Push Topic **events** to subscribers
  - Subscription can filter by event payload
- Azure Service Bus – "Full suite for business messages!"
  - Queue or Topics
  - FIFO ordering, transactions, fault handling
  - Sessions, dead-letter queues, policies, duplicate detection
- Event Hubs – "Ordered event streaming"
  - Beware of racing conditions

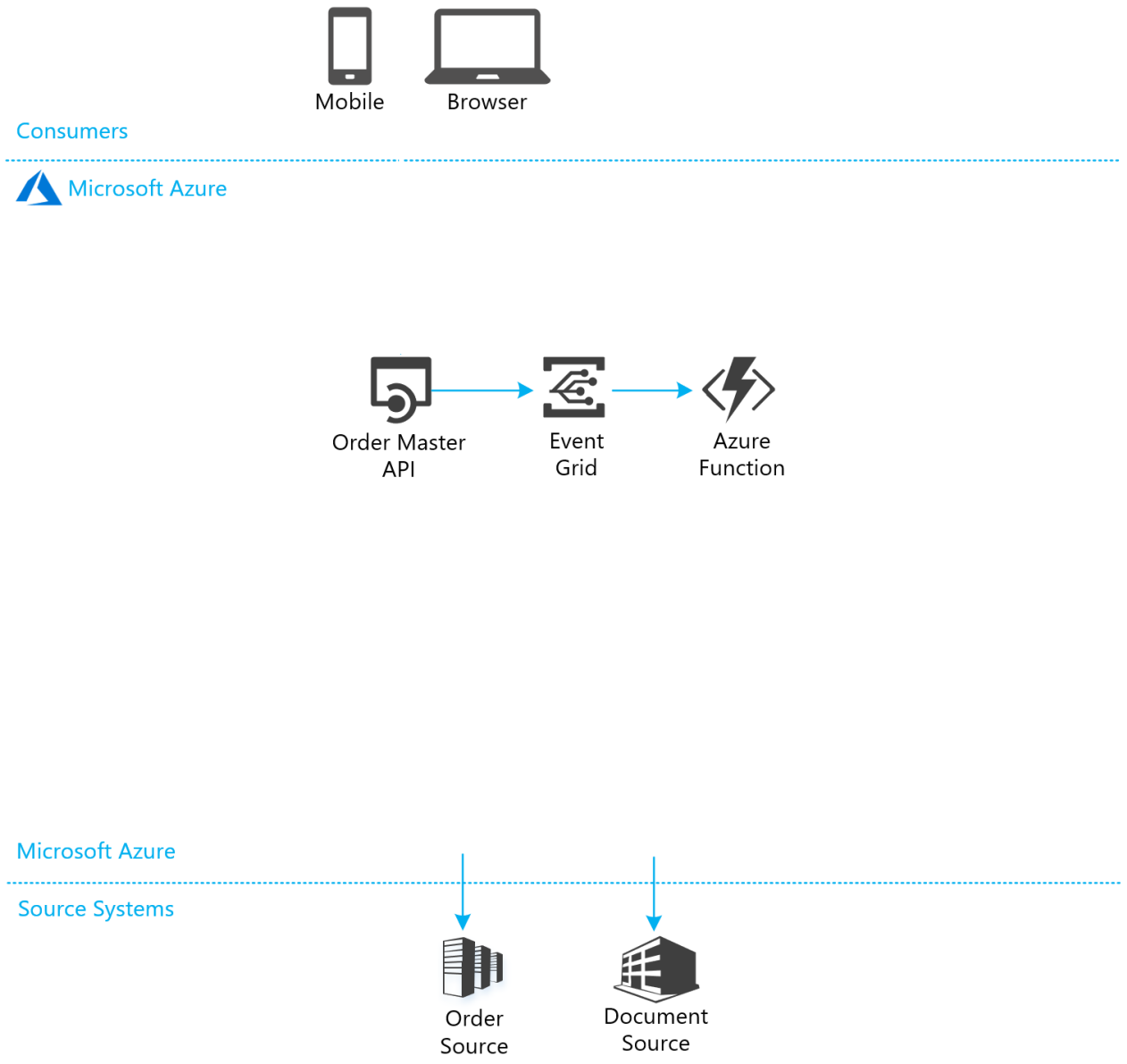


‘Order Master’ Concept



Features

- Users: sales and clients
- Users search for purchase orders in free text
- There are millions of meaningfully diverse orders
- Orders also
  - Refer to sales data: accounts, contacts, activities, etc.
  - Contain binary objects: PO documents and images
- Report generation is compute intensive



Architecture

Azure Event Grid for eventing and messaging

- Order Master API sends 'Command' events to Azure Function subscriber-processor
- It is also likely that Order Source's Azure Function notifies other components of updates via Event Grid; e.g. for the API to be aware of cache invalidation requests etc

Azure API Management is in use by the corporation

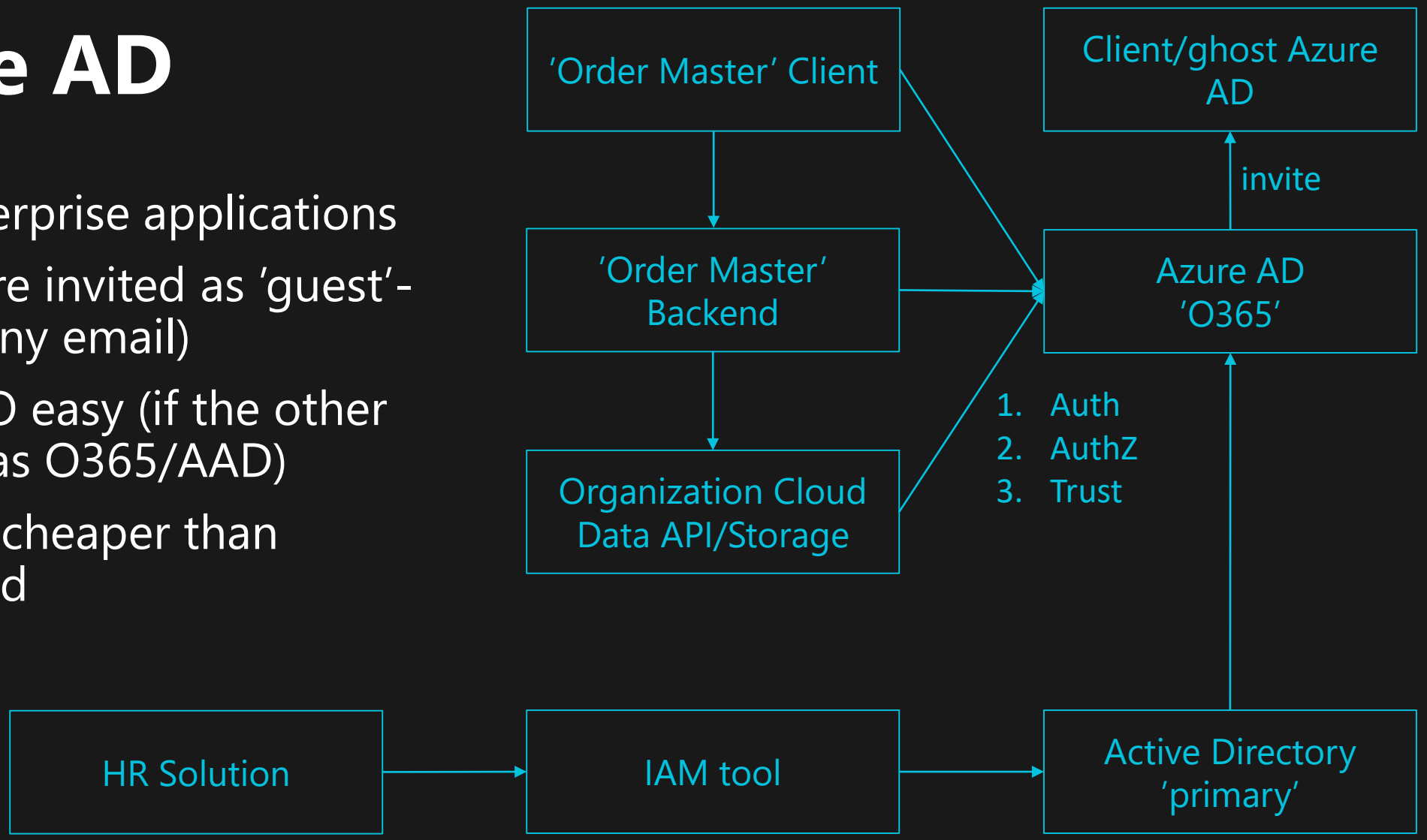
Naturally we route the requests from consumer clients via the API Management

Application

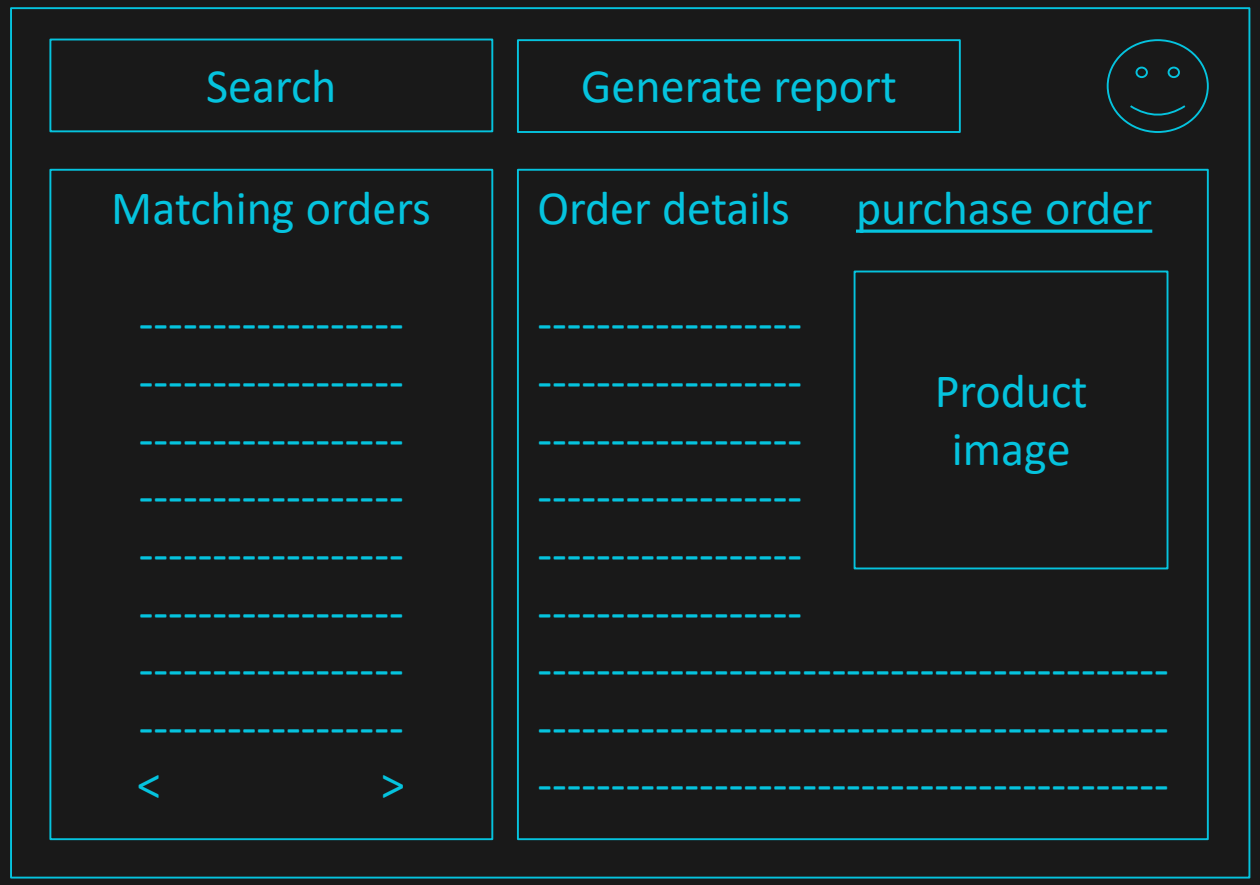
Monitoring	Logging, debug data, time series, queryable, custom events, dashboards
Identity	Externals, internals, governance, regulations, SSO, protocols, current IAM
Compute	Control, platform lock-in, infrastructure, scaling, architecture, management
Messaging	Queue, publish/subscribe, routing, persistence, ordering, security, sessions
Storages	Relational, binary, unstructured, caching, analytics, search

Azure AD

- For enterprise applications
- Users are invited as 'guest'-users (any email)
- B2B SSO easy (if the other party has O365/AAD)
- Usually cheaper than expected

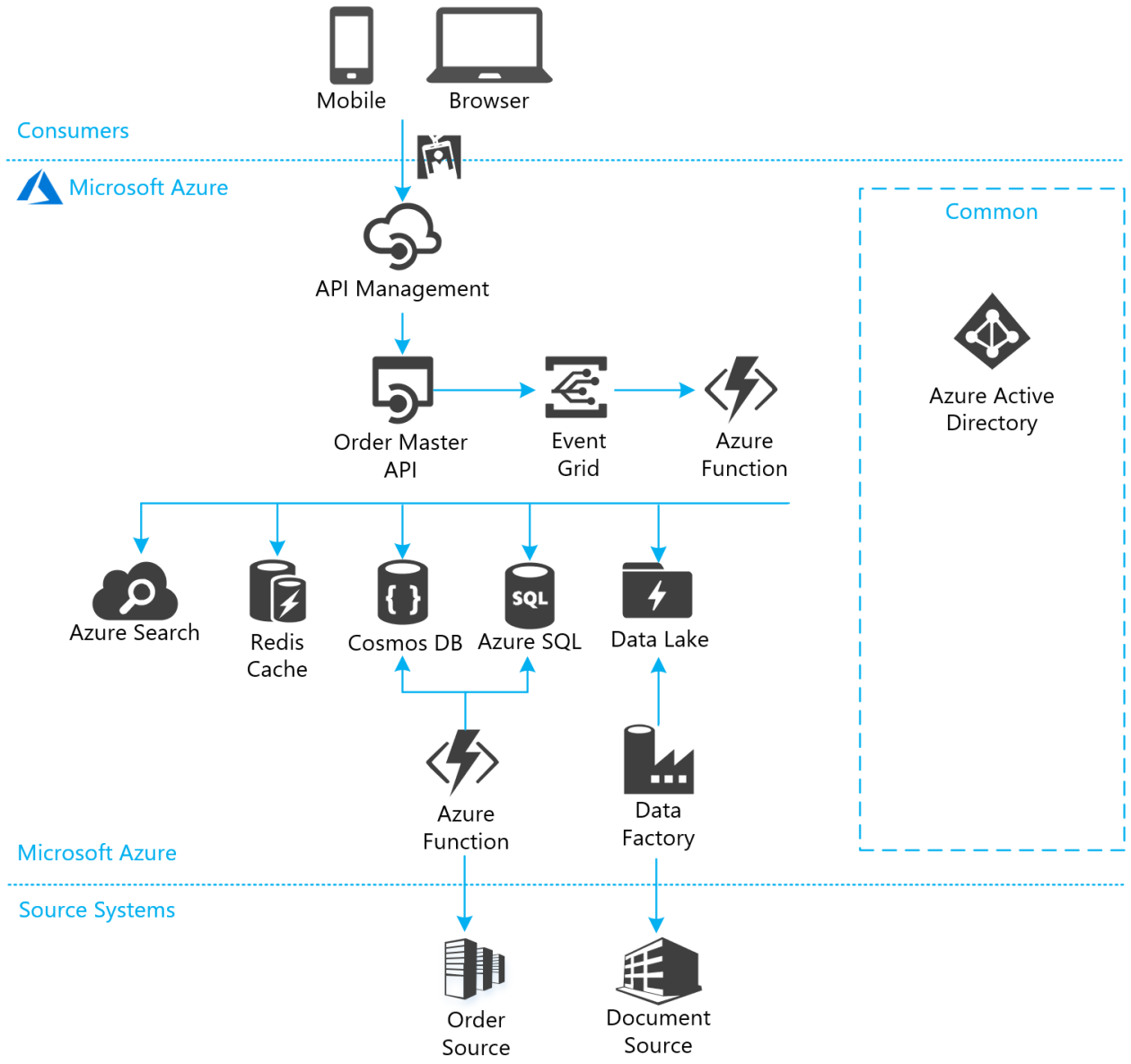


‘Order Master’ Concept



Features

- Users: sales and clients
- Users search for purchase orders in free text
- There are millions of meaningfully diverse orders
- Orders also
  - Refer to sales data: accounts, contacts, activities, etc.
  - Contain binary objects: PO documents and images
- Report generation is compute intensive



Architecture

Azure AD handles authentication and authorization*

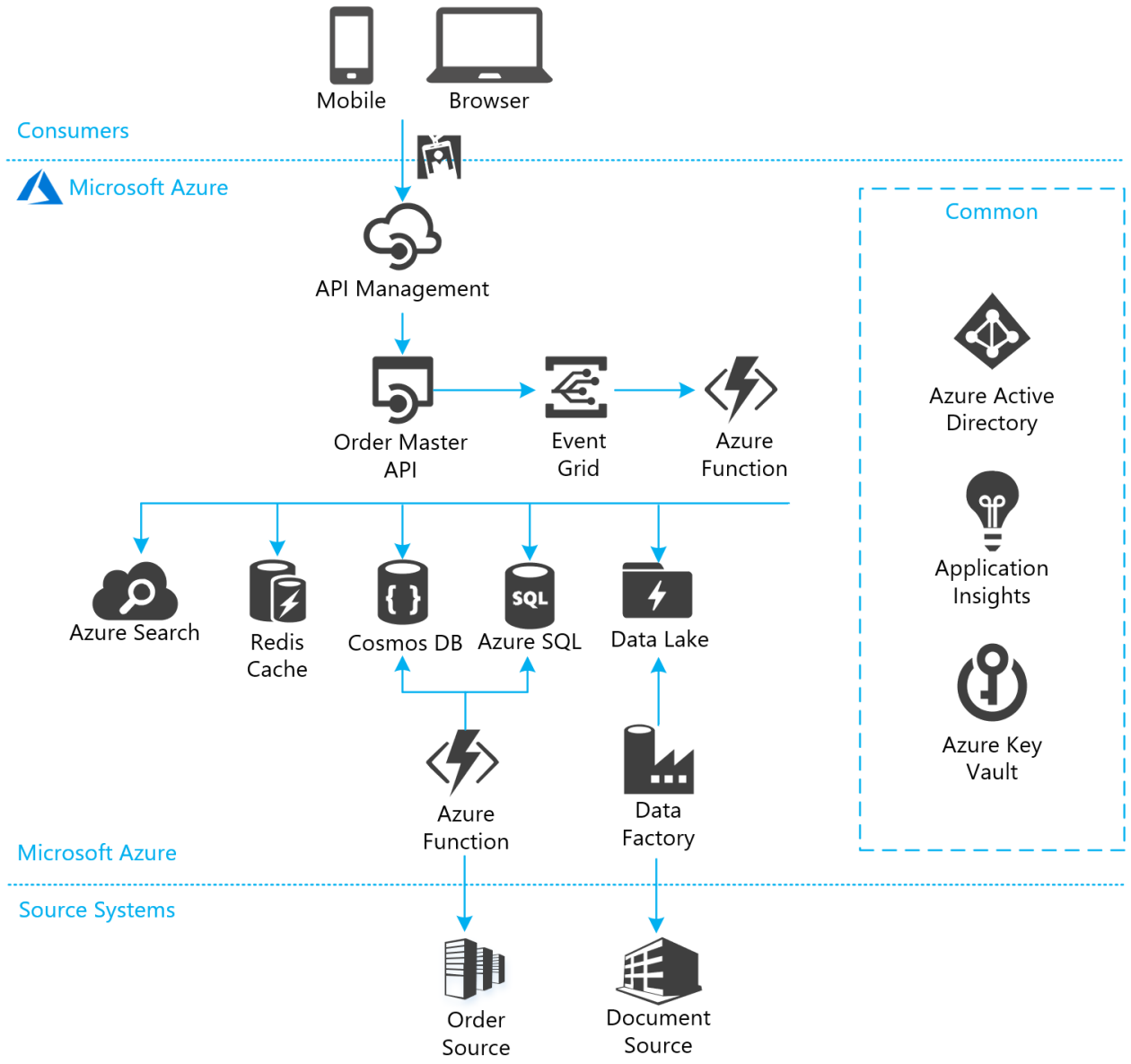
- Applications trust each other via AAD configuration
- User is authentication and authorized down to database layer as a user
- *Note: Authorization schemes often require additional metadata that is outside the AAD

Application

Monitoring	Logging, debug data, time series, queryable, custom events, dashboards
Identity	Externals, internals, governance, regulations, SSO, protocols, current IAM
Compute	Control, platform lock-in, infrastructure, scaling, architecture, management
Messaging	Queue, publish/subscribe, routing, persistence, ordering, security, sessions
Storages	Relational, binary, unstructured, caching, analytics, search

Monitoring

- Application Insights
  - Analytics with Kusto query language
  - Client-side support as well
- Azure Monitor
  - Unified view for Azure PaaS products
- Dashboard
  - Custom events and metrics
- Event hubs
  - Azure Data Explorer (Kusto)
- API Management
  - API monitoring



Architecture

Application Insights contains the health telemetry

- With custom events and metrics; for debugging purposes

Azure Key Vault takes care of secrets

- Certificates, connection strings, etc.

Take-aways

- Concept first, architecture second
- Storage can be the hardest thing to change later
- In Compute, PaaS + FaaS = the least amount of admin
- Wrong messaging choices can halt your app, easy
- PaaS drives platform lock-in, so evaluate your ROI
- On Identity, try and let someone else take care of your user store
- Monitoring is never good enough on launch day, try and improve it during the feature freeze period before production release
- At the end of the day - it’s just talking to people and coding ☺

ZURE