App modernization by using Dapr and Azure Container Apps

Sergii Bielskyi

ABOUT MYSELF

Current: Head of R&D at ELEKS

Previous: Cloud Solution Architect

→ E-doc at APU

→ MCS at MS Ukraine

MVP (8 times), Microsoft Azure and IoT

Lead of IoT community

IoT community https://www.facebook.com/groups/iot.ua

Facebook https://www.facebook.com/sergey.belskiy

Mastodon https://techhub.social/@sergiib

Blog https://medium.com/@sergiibielskyi

LinkedIn https://www.linkedin.com/in/sergiib

ELEKS GLOBAL PRESENCE

18

offices

2,100+

global headcount

1450+

delivery experts

Americas

Chicago, USA
Las Vegas, USA
Toronto, Canada

Europe

Tallinn, Estonia
Berlin, Germany
London, UK
Rzeszow, Poland
Krakow, Poland
Zürich, Switzerland
Split, Croatia
Zagreb, Croatia

Ukraine

Lviv
Kyiv
Ternopil
Ivano Frankivsk

Middle East

Ajman, UAE
Al Jubail, KSA

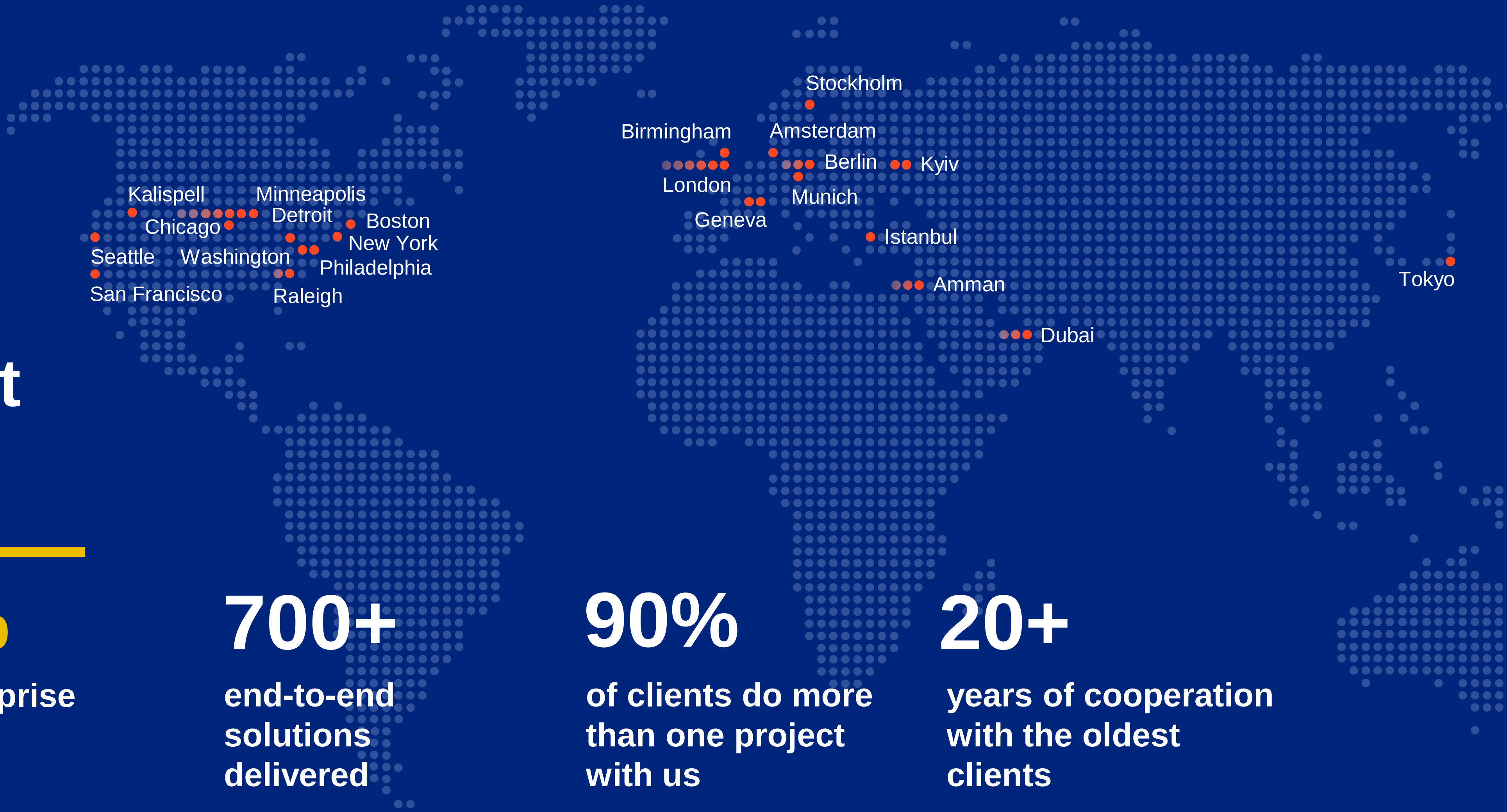
Asia

Tokyo, JAPAN

Headquarters ●
Development Centres ●
Partners ●
International Offices ●
Affiliates ●

eleks

MEET OUR CLIENTS



150+
active client accounts

75%
for SMB

25%
for Enterprise

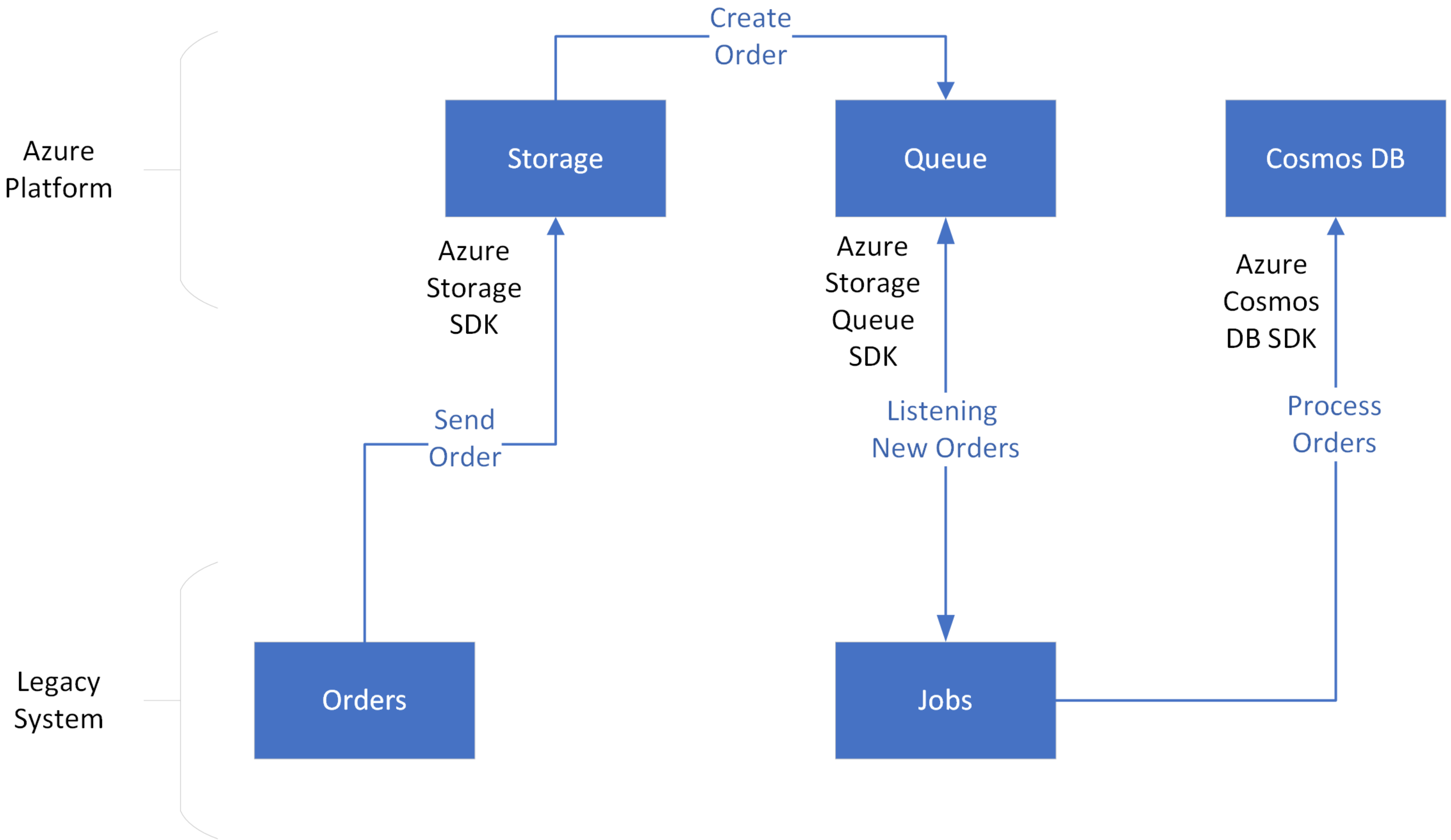
700+
end-to-end solutions delivered

90%
of clients do more than one project with us

20+
years of cooperation with the oldest clients

eleks

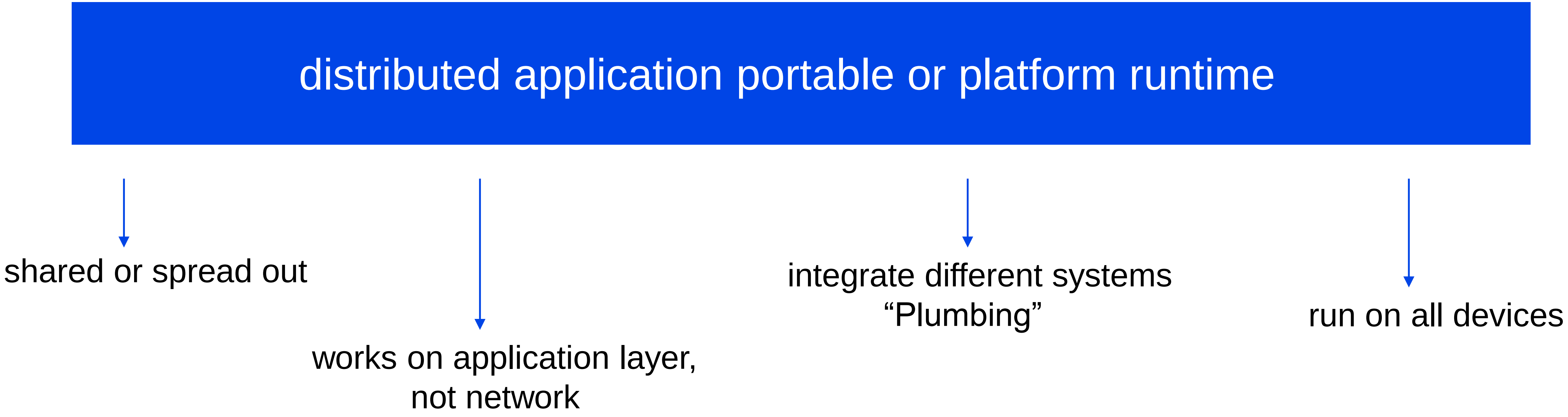
STORY ABOUT EXISTING LEGACY SYSTEM



Why we need to modernize it

- “Everybody do it”
- Hard to deliver new features or application changes
- Dependencies of hosting platform / OS
- Support maintenance of new version of SDKs
- Dependencies of SDKs
- Technical knowledge of knowing different specific SDKs

Why DAPR



distributed application portable or platform runtime

shared or spread out

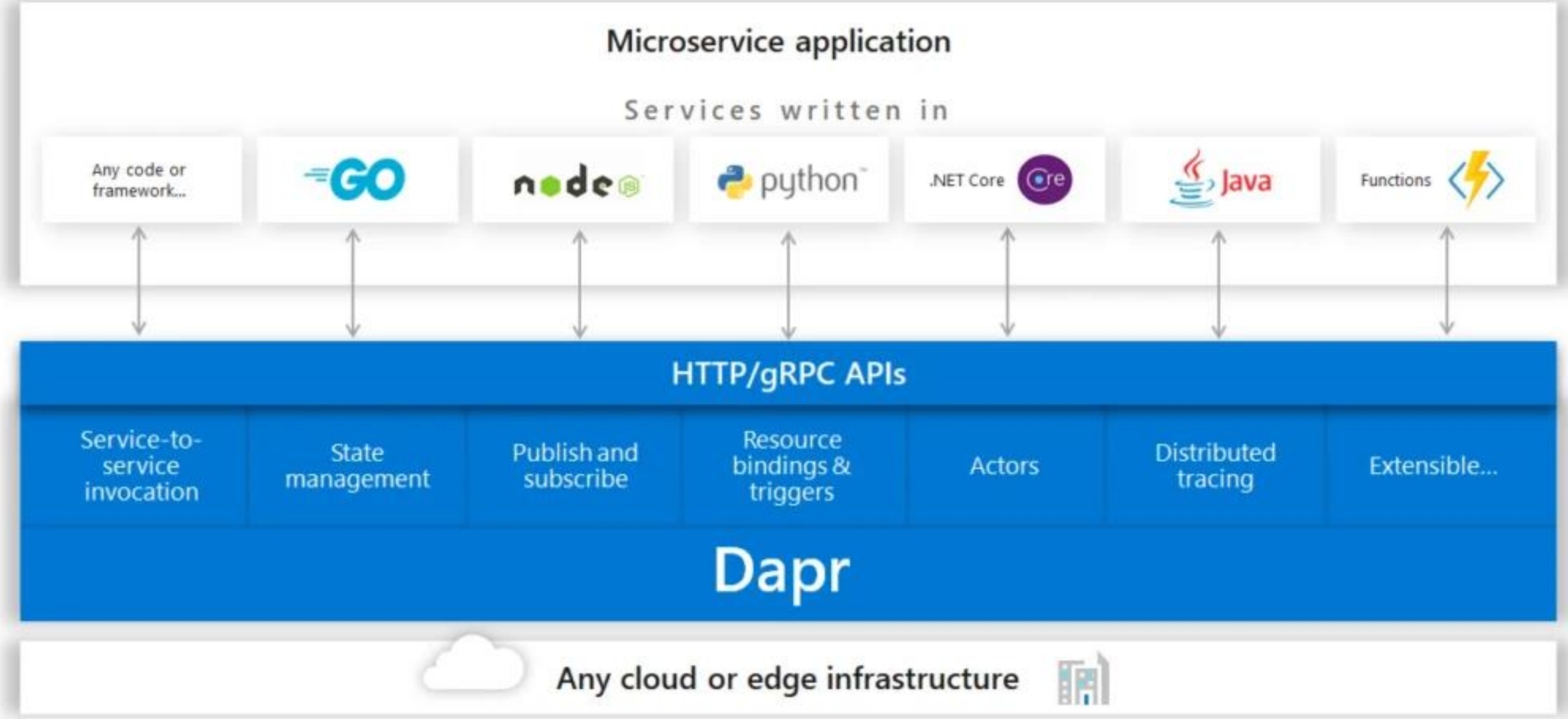
works on application layer,
not network

integrate different systems
“Plumbing”

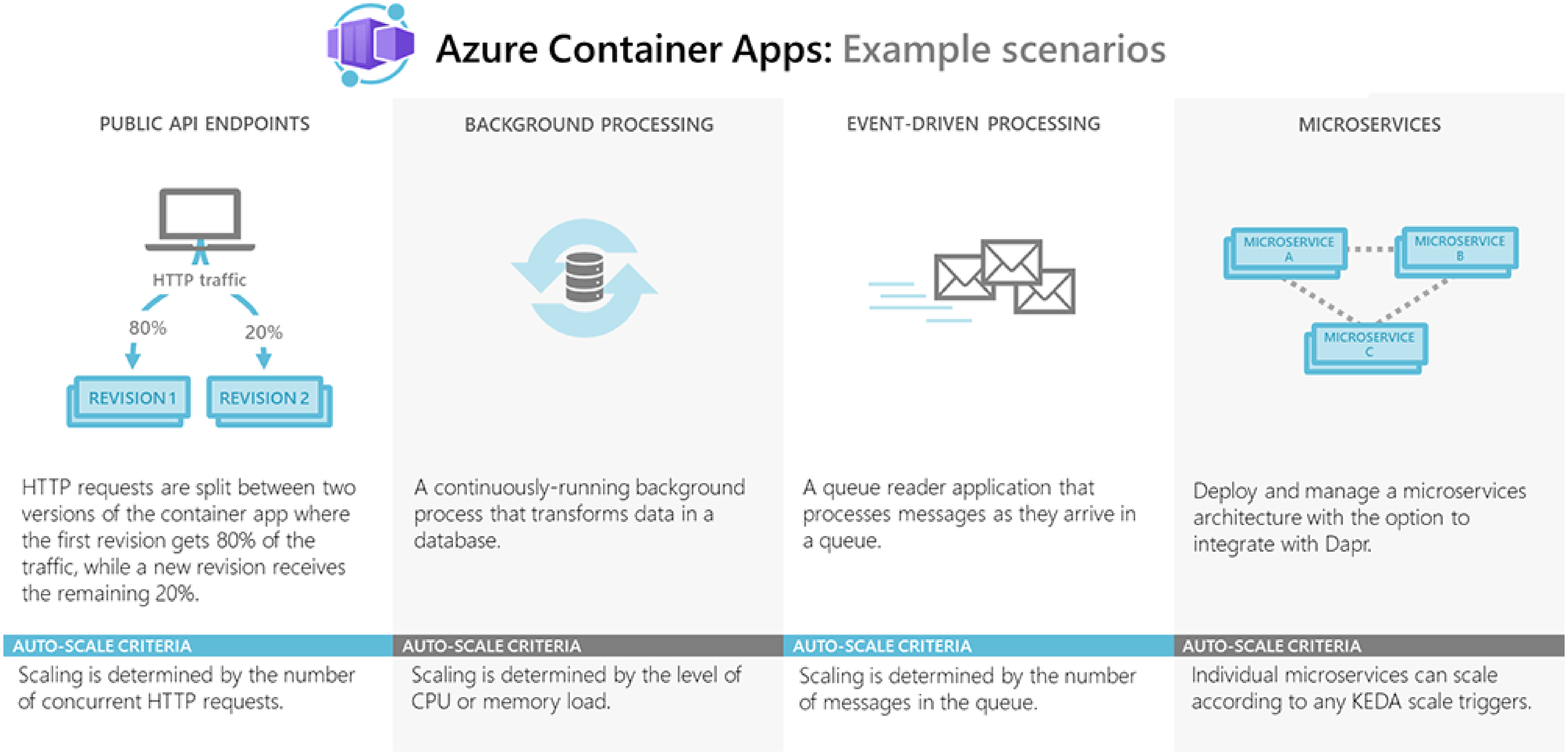
run on all devices

DARP IS

https://docs.dapr.io



WHY AZURE CONTAINER APPS



Azure Container Apps: Example scenarios

PUBLIC API ENDPOINTS

HTTP requests are split between two versions of the container app where the first revision gets 80% of the traffic, while a new revision receives the remaining 20%.

AUTO-SCALE CRITERIA

Scaling is determined by the number of concurrent HTTP requests.

BACKGROUND PROCESSING

A continuously-running background process that transforms data in a database.

AUTO-SCALE CRITERIA

Scaling is determined by the level of CPU or memory load.

EVENT-DRIVEN PROCESSING

A queue reader application that processes messages as they arrive in a queue.

AUTO-SCALE CRITERIA

Scaling is determined by the number of messages in the queue.

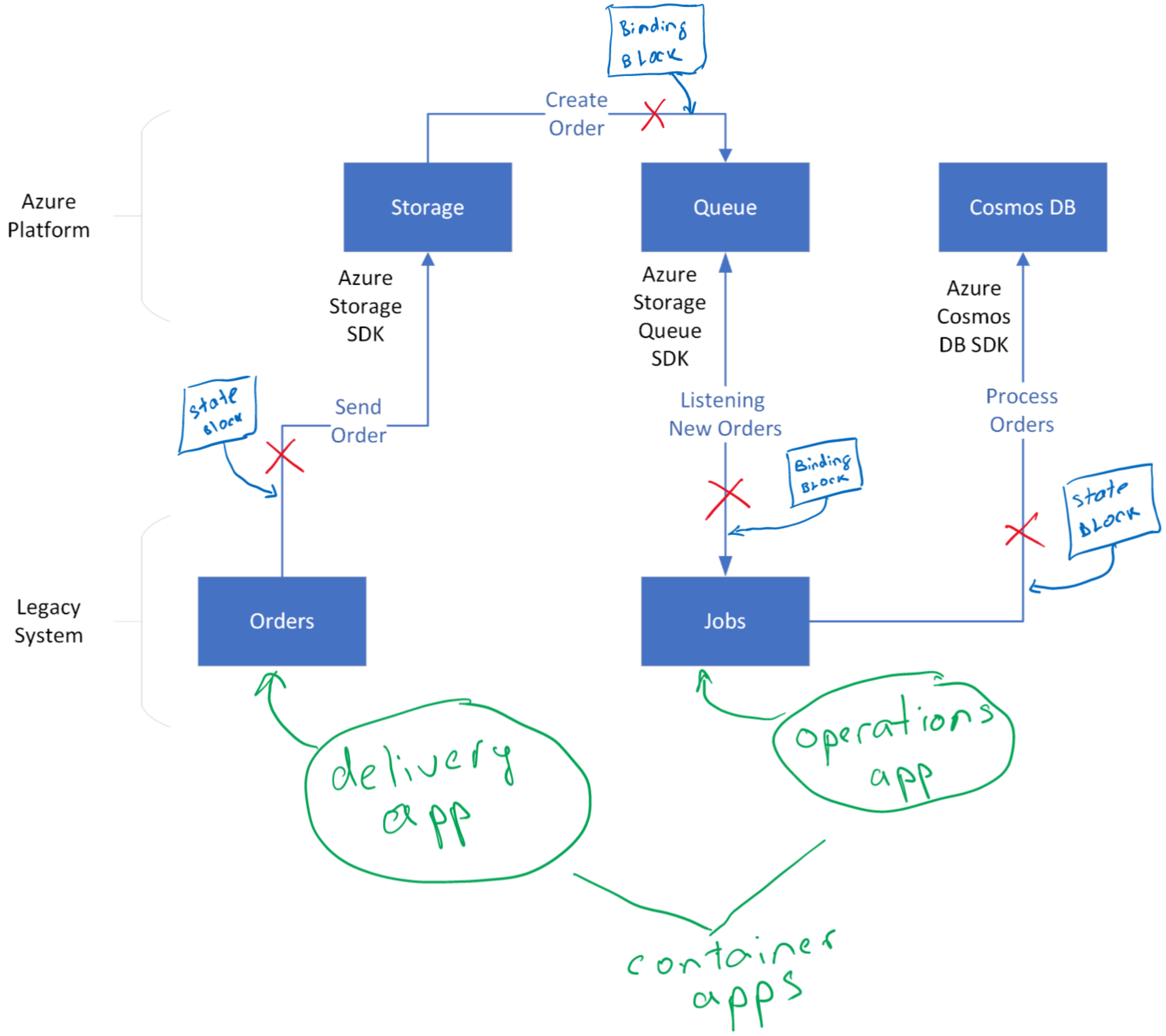
MICROSERVICES

Deploy and manage a microservices architecture with the option to integrate with Dapr.

AUTO-SCALE CRITERIA

Individual microservices can scale according to any KEDA scale triggers.

NEW ARCHITECTURE VISION



Dapr building blocks

→ State management
- Azure Storage Container
- Azure Cosmos DB
- Azure PostgreSQL

→ Binding input/output
- Azure Storage Queue

→ Secrets
- Azure Key Vault

→ Azure Container Apps
- Delivery app
- Operations app

HOW TO START WITH DAPR

→ To install Dapr engine (3 components included) cli - dapr init

→ Preparation of Dapr components (Dapr blocks)

→ To install Dapr Client SDK from nuget ([NuGet Gallery | Dapr.Client 1.9.0])

→ Injection into the code by using Dapr SDK

→ Run Dapr application

```
--Start dapr apps

dapr run -a deliveryapp -p 60000 -d
components -- dotnet run



--start dapr apps with input bindings

dapr run -a checkoutapp -p 50000 -d
components -- dotnet run --urls
http://*:50000
```

HOW TO START USING AZURE CONTAINER APPS

→ Preparation of container environment services in Azure

→ To build docker containers (`-r ubuntu.18.04-x64`)

- Any Linux-based x86-64 (linux/amd64) container image
- Containers from any public or private container registry

→ To publish docker to ACR

→ Preparation of bicep scripts to deploy azure container apps

DEMO

Demonstrate components of legacy system

Cloud resources

Running locally without Dapr and with Dapr

Change/deploy delivery part to use Dapr

Bicep scripts

Docker file

New controller

Dapr components

Change/deploy operations part to use Dapr

Bicep scripts

Docker file

New controller

Dapr components

Setup CI/CD with github activities

Configure operations part to use github

Change code path

Change state management for operations app

Via cloud bash console