Components & services reuse with JAMstack, Azure & Microsoft Graph

JAMstack is cool for me because of REUSABILITY!

Reuse Javascript

Reuse API

Reuse Markup

Why is Reusability important?

Developer Experience

Developer Velocity

Quality Economy

APIs and Cloud Services

Reusable APIs, accessed over HTTPS with JavaScript
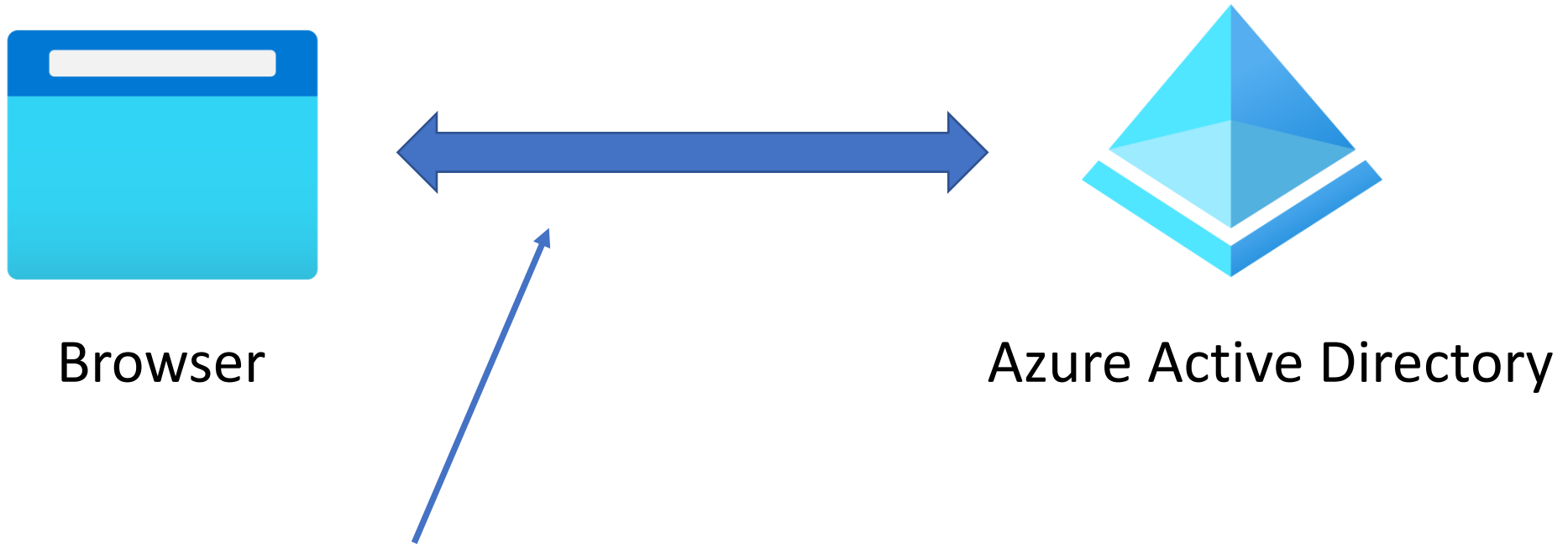
Cloud services

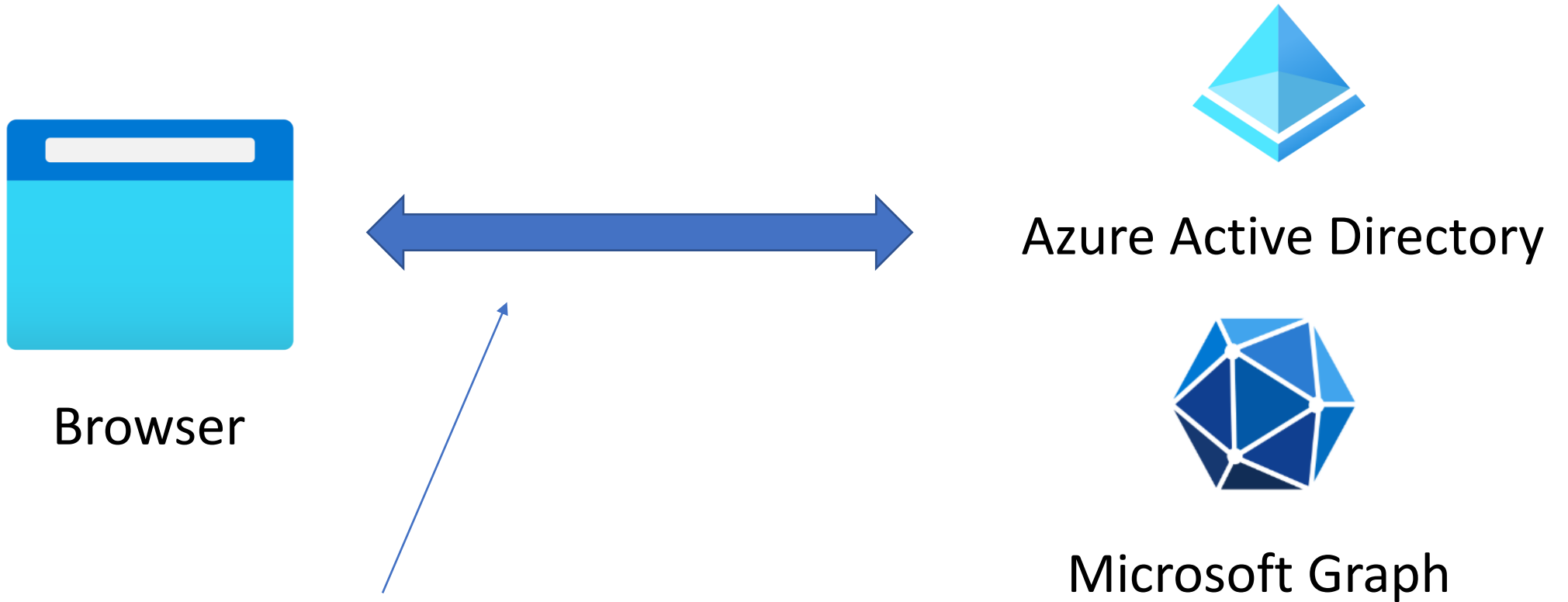In-house APIs

Authentication with Azure Active Directory
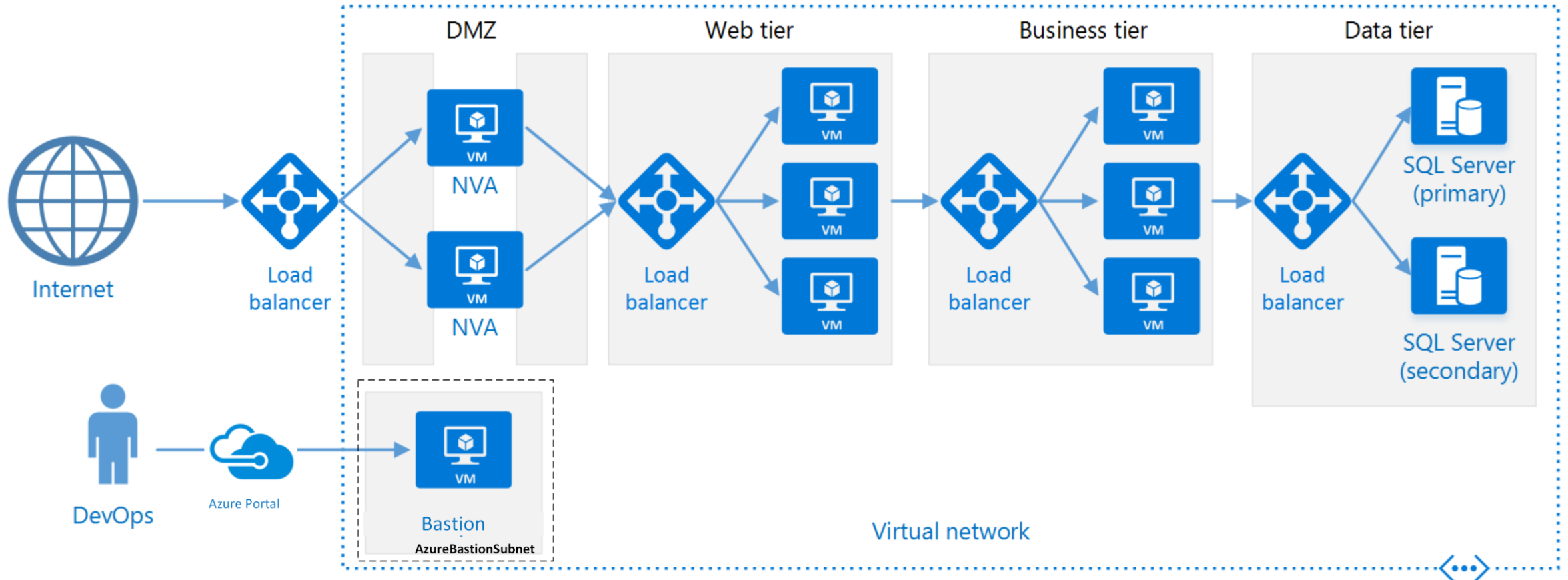
Browser

Azure Active Directory

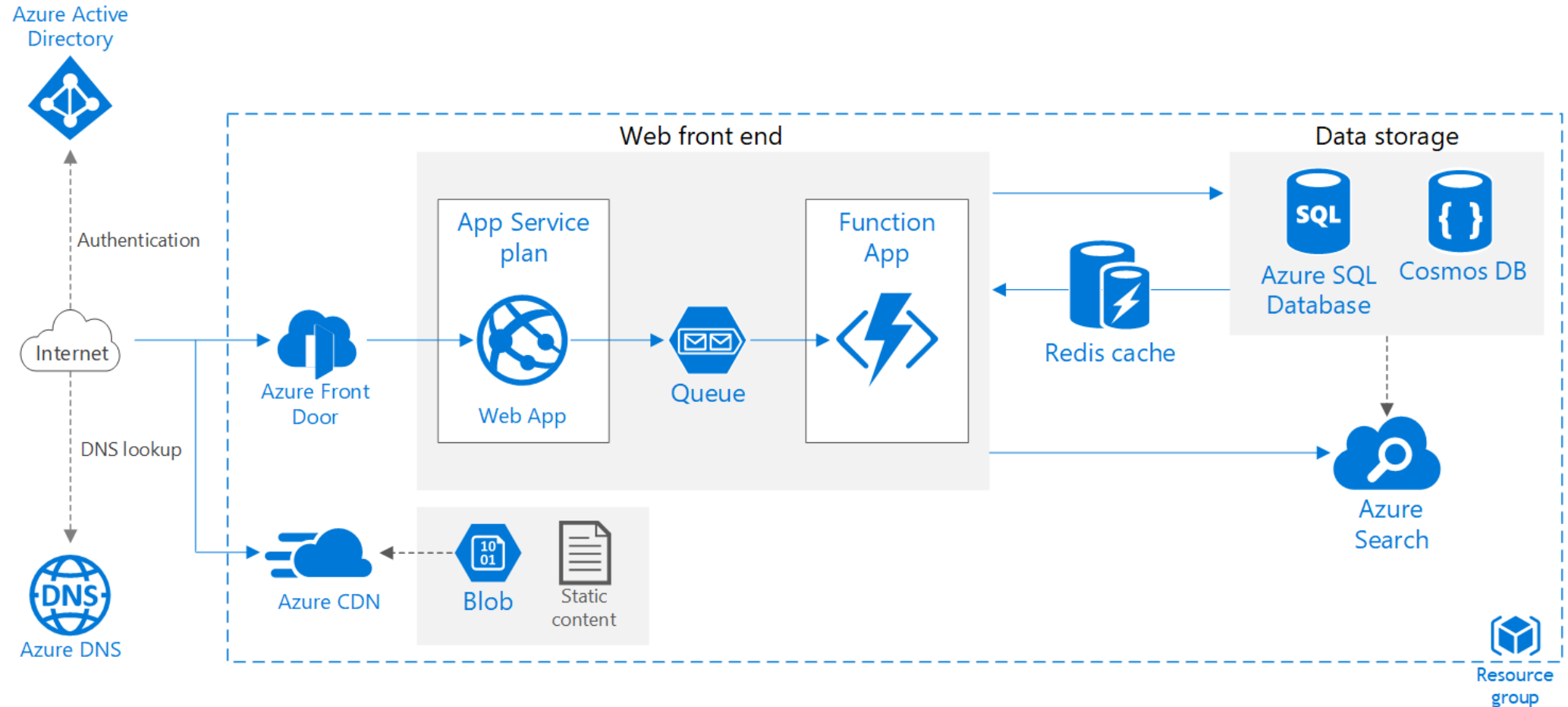No application servers!

Authentication with Azure Active Directory

Traditional architecture

Scalable web application architecture

On Premises	Infrastructure as a Service	Platform as a Service	Software as a Service
Applications	Applications	Applications	Applications
Data	Data	Data	Data
Runtime	Runtime	Runtime	Runtime
Middleware	Middleware	Middleware	Middleware
O/S	O/S	O/S	O/S
Virtualization	Virtualization	Virtualization	Virtualization
Servers	Servers	Servers	Servers
Storage	Storage	Storage	Storage
Networking	Networking	Networking	Networking

Managed by in-house teams

Managed Service (Cloud)

L	Linux	Operating System
A	Apache	Web Server
M	MySQL	Database
P	PHP	Application

J	JavaScript	Application
A	API	Application
M	Markup	Application

Infrastructure & control   Abstraction & reuse
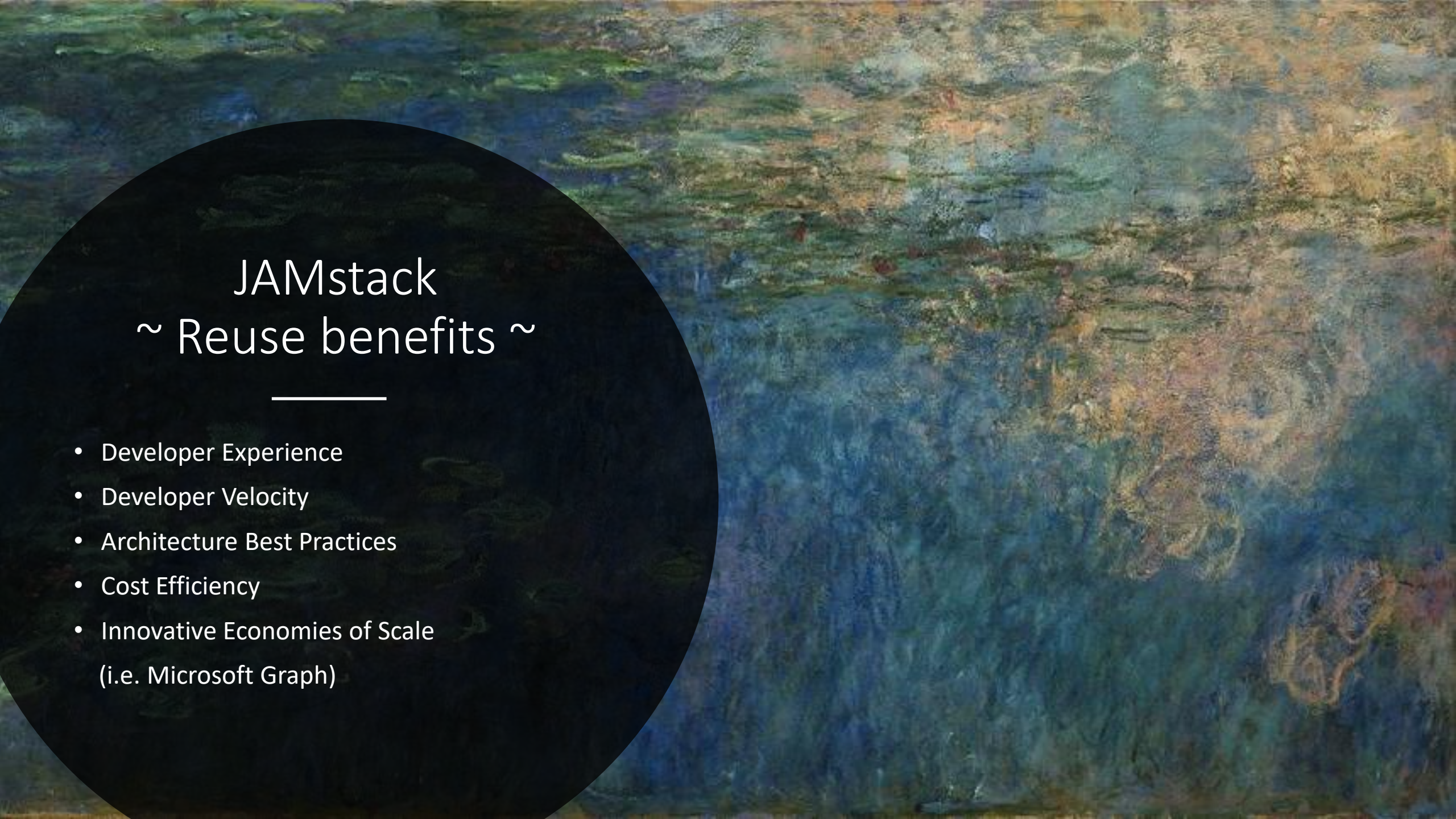
Operations   Applications

Single Page Applications (SPAs)
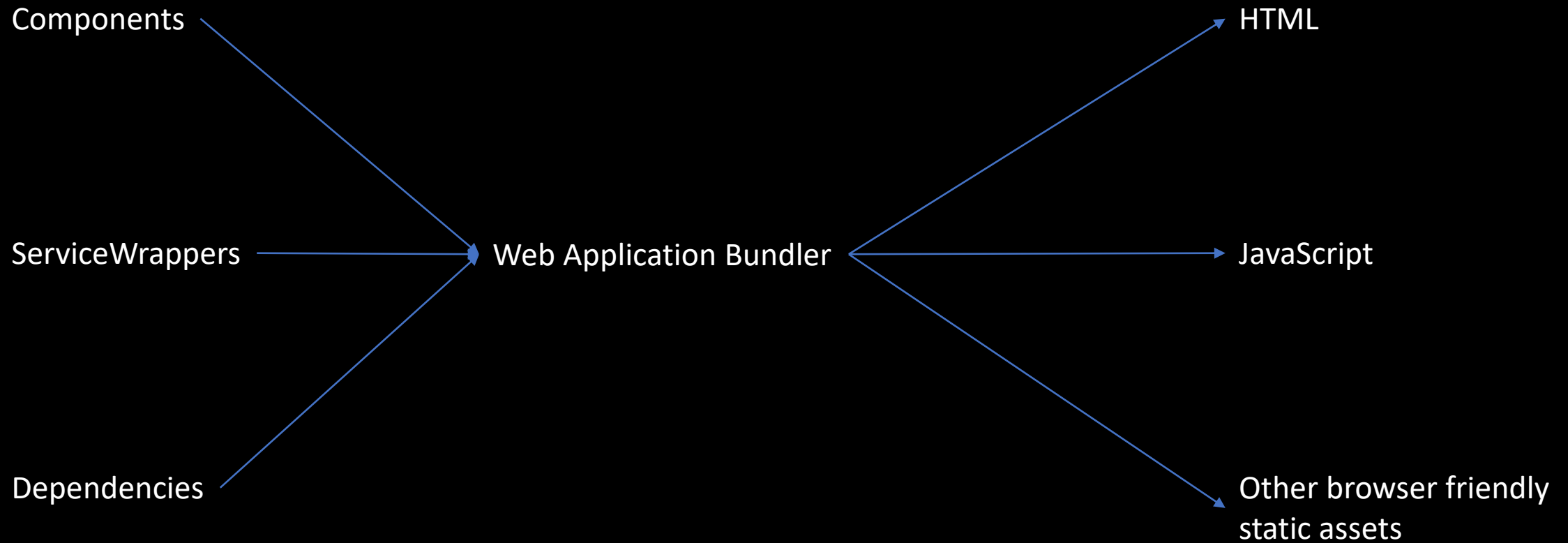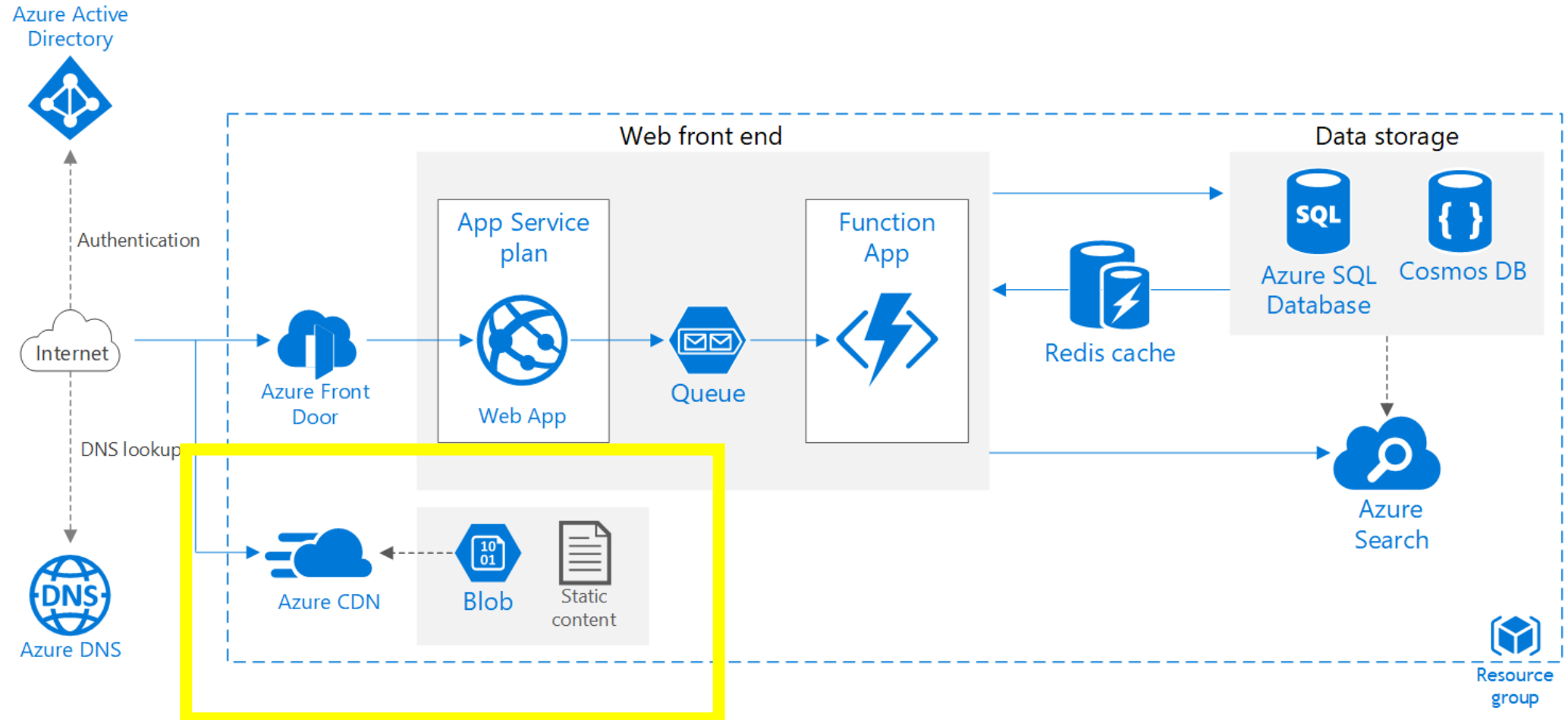
Components

ServiceWrappers

Dependencies

Web Application Bundler

HTML

JavaScript

Other browser friendly static assets

Scalable web application architecture

JAMstack
~ Reuse benefits ~
———

- Developer Experience

- Developer Velocity

- Architecture Best Practices

- Cost Efficiency

- Innovative Economies of Scale

  (i.e. Microsoft Graph)