

DotNet 2021

ONLINE TECH CONFERENCE

22nd June 2021

#DotNet2021

.NET 6 Cloud Native



www.dotnet2021.com

DotNet 2021

ORGANIZATION

plain
concepts 

IN COOPERATION WITH

FUNDACIÓN
GOMAESPUMA
"Educando con una sonrisa."

SPONSORS

 Microsoft

DevsDNA 

 intelequia

 My Public
Inbox

#DotNet2021



Glenn Condron

Principal Program Manager

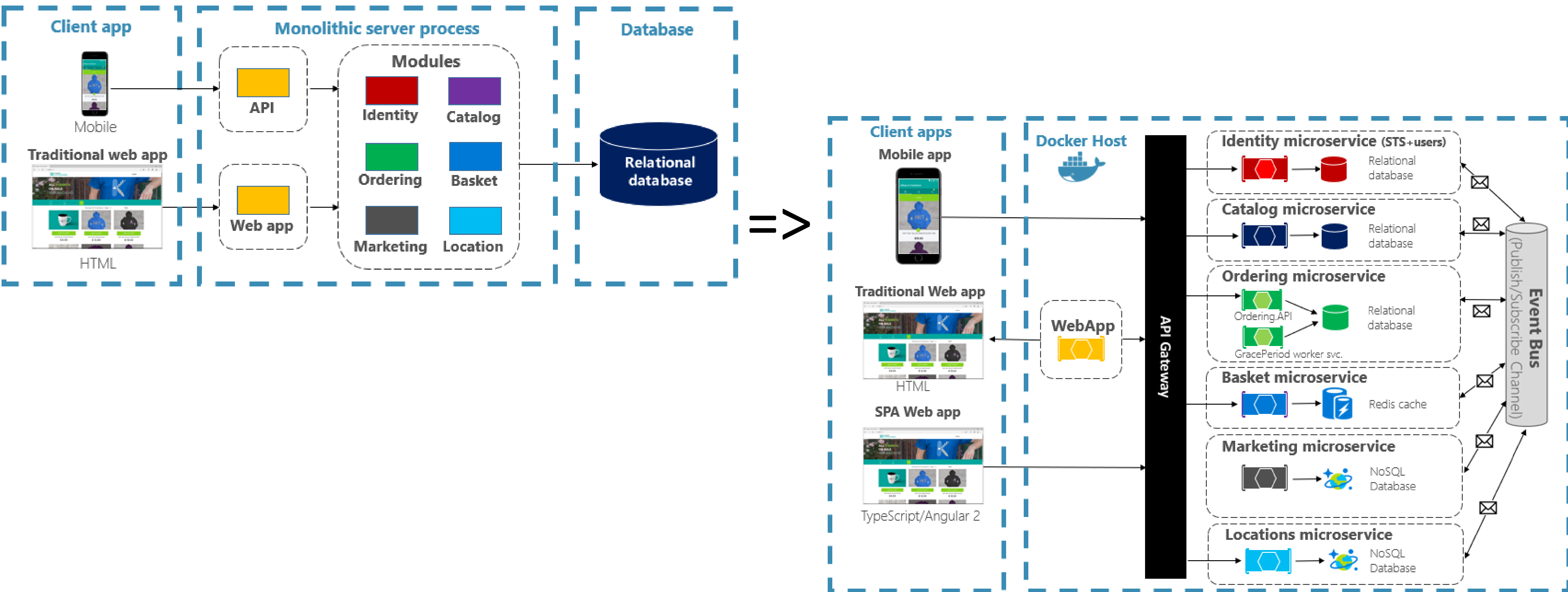
I am a PM on the .NET team at Microsoft and have worked on .NET Core from the beginning.

I am the manager of a PM team that are responsible for a whole bunch of .NET app models, most relevant to this talks is that the team builds ASP.NET Core.

@condrong

glennc@microsoft.com

Cloud Native



What is Cloud Native?

Isn't it just all buzz words that don't mean anything?

- First started using it to mean the opposite of lift and shift, i.e. an app/solution that was built with the cloud in mind.
 - 12 factor
 - Embrace failure
 - SPA (expensive compute cheap storage)

What is Cloud Native?

Somehow turned into this



CLOUD NATIVE TRAIL MAP

The Cloud Native Landscape (CNCFL) has a large number of options. This Cloud Native Trail Map is a recommended process for leveraging open source, cloud native technologies. At each step, you can choose a vendor-supported offering or do it yourself, and everything after step #3 is optional based on your circumstances.

HELP ALONG THE WAY

A. Training and Certification

Consider training offerings from CNCF and then take the exam to become a Certified Kubernetes Administrator or a Certified Kubernetes Application Developer [cncf.io/training](https://www.cncf.io/training)

B. Consulting Help

If you want assistance with Kubernetes and the surrounding ecosystem, consider leveraging a Kubernetes Certified Service Provider [cncf.io/csp/](https://www.cncf.io/csp/)

C. Join CNCF's End User Community

For companies that don't offer cloud native services externally [cncf.io/end-user/](https://www.cncf.io/end-user/)

WHAT IS CLOUD NATIVE?

Cloud native technologies empower organizations to build resilient and scalable applications in modern, dynamic environments such as public, private, and hybrid clouds. Containers, service meshes, microservices, immutable infrastructure, and declarative APIs exemplify this approach.

These techniques enable loosely coupled systems that are resilient, manageable, and observable. Combined with robust automation, they allow engineers to make high-impact changes frequently and predictably with minimal toil.

The Cloud Native Computing Foundation seeks to drive adoption of this paradigm by fostering and sustaining an ecosystem of open source, vendor-neutral projects. We democratize state-of-the-art patterns to make these innovations accessible for everyone.

l.cncf.io

v20200501



CNCF Cloud Native Interactive Landscape

The Cloud Native Trail Map (png, pdf) is CNCF's recommended path through the cloud native landscape. The cloud native landscape (png, pdf), serverless landscape (png, pdf), and member landscape (png, pdf) are dynamically generated below. Please open a pull request to correct any issues. Greyed logos are not open source. Last Update: 15:09:09Z

You are viewing 944 cards with a total of 2,640,178 stars, market cap of \$15.24T and funding of \$16.62B.

Landscapes: Landscape, Card Mode, Serverless, Members

Categories: Database, Streaming & Messaging, Application Definition & Image Build, Continuous Integration & Delivery, Platform, Certified Kubernetes - Distribution, Certified Kubernetes - Hosted, Certified Kubernetes - Installer, PaaS/Container Service, Observability and Analysis, Monitoring, Logging, Tracing, Chaos Engineering, Scheduling & Orchestration, Coordination & Service Discovery, Remote Procedure Call, Service Proxy, API Gateway, Service Mesh, Cloud Native Storage, Container Runtime, Cloud Native Network, Automation & Configuration, Container Registry, Security & Compliance, Key Management, Kubernetes Certified Service Provider, Kubernetes Training Partner

Cloud Native

Why? How?

- **Resilience and Scalability:** Utilizing the cloud to reduce risk of outages or availability
- **Efficiency:** Architect for the cloud to get the most efficient, cheapest to run, app
- **Velocity:** Faster conversion of ideas to code

Cloud Native

Why? How?

- **Resilience and Scalability:** Utilizing the cloud to reduce risk of outages or availability
- **Efficiency:** Architect for the cloud to get the most efficient, cheapest to run, app
- **Velocity:** Faster conversion of ideas to code

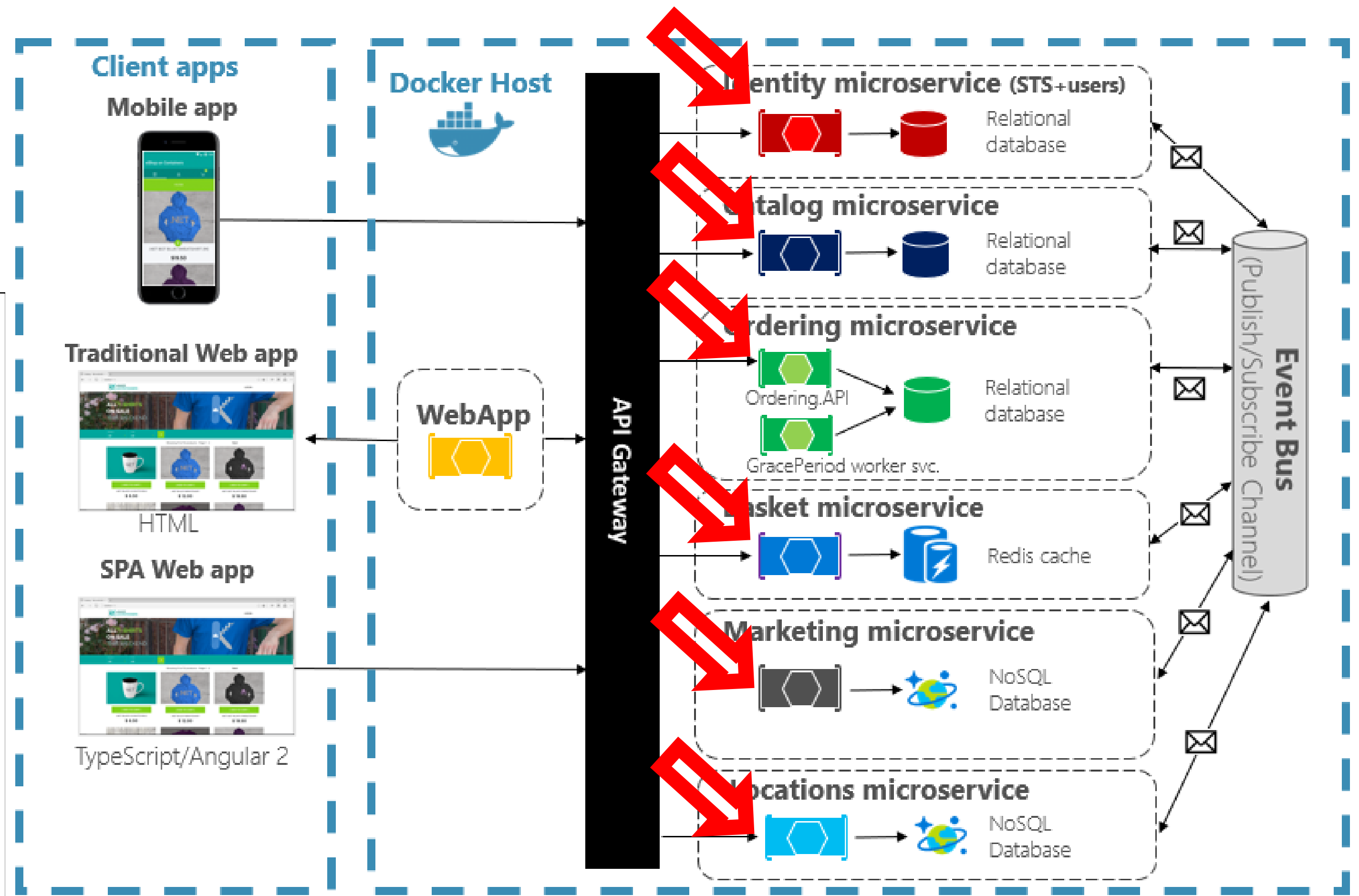
- If you want all these things, then technology seemingly unrelated to the cloud becomes critical such as CI/CD.

- When talking about Cloud Native and .NET we are generally talking about things in .NET that make it easier/better to do the kinds of things people do when they are trying to achieve all of these. Microservices, Containers, Kubernetes, Observability

Cloud Native

APIs

- Minimal hosting and map for more concise services
- gRPC load balancing, better load balancing in k8s clusters.
- Streaming JSON serialization
- Lots of work on the common side coming later



Minimal Apps

```
var builder = WebApplication.CreateBuilder(args);
await using var app = builder.Build();

if (app.Environment.IsDevelopment())
{
    app.UseDeveloperExceptionPage();
}

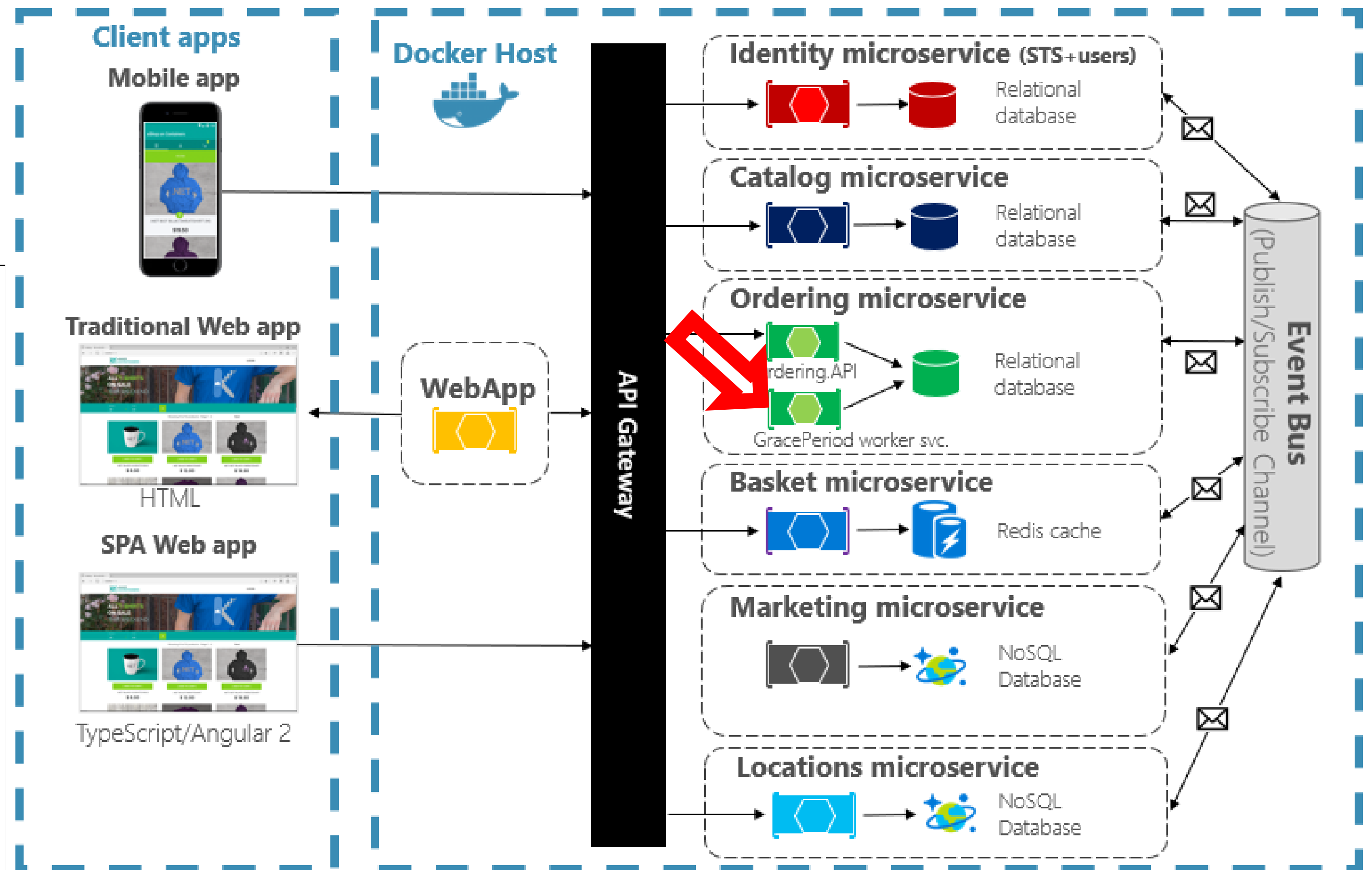
app.MapGet("/", () => "Hello World!");

await app.RunAsync();
```

Cloud Native

Workers

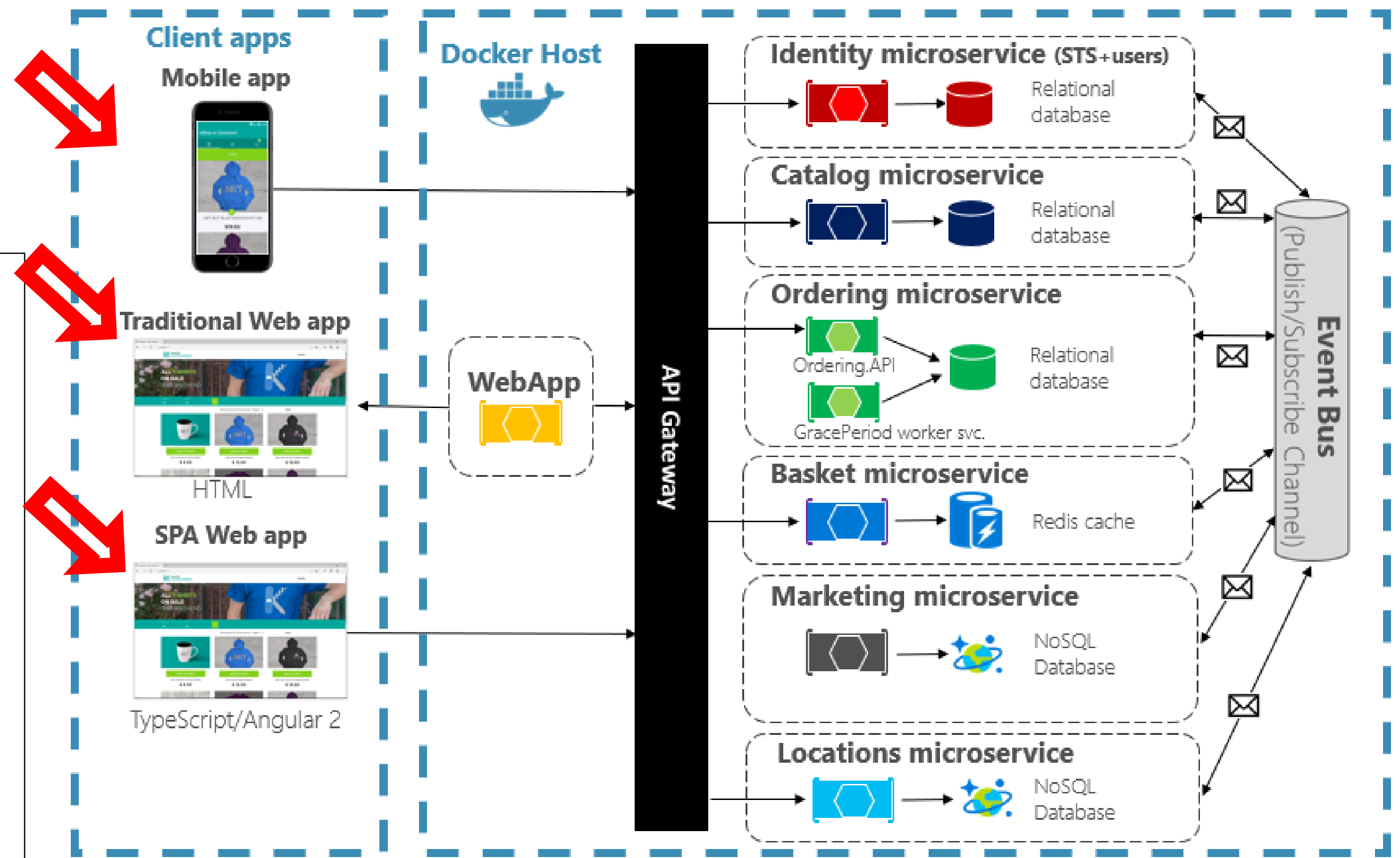
- Not a lot of explicit work here but I want to call out that the template exists and we expect to be doing some work here in the not too distant future.



Cloud Native

Client Apps

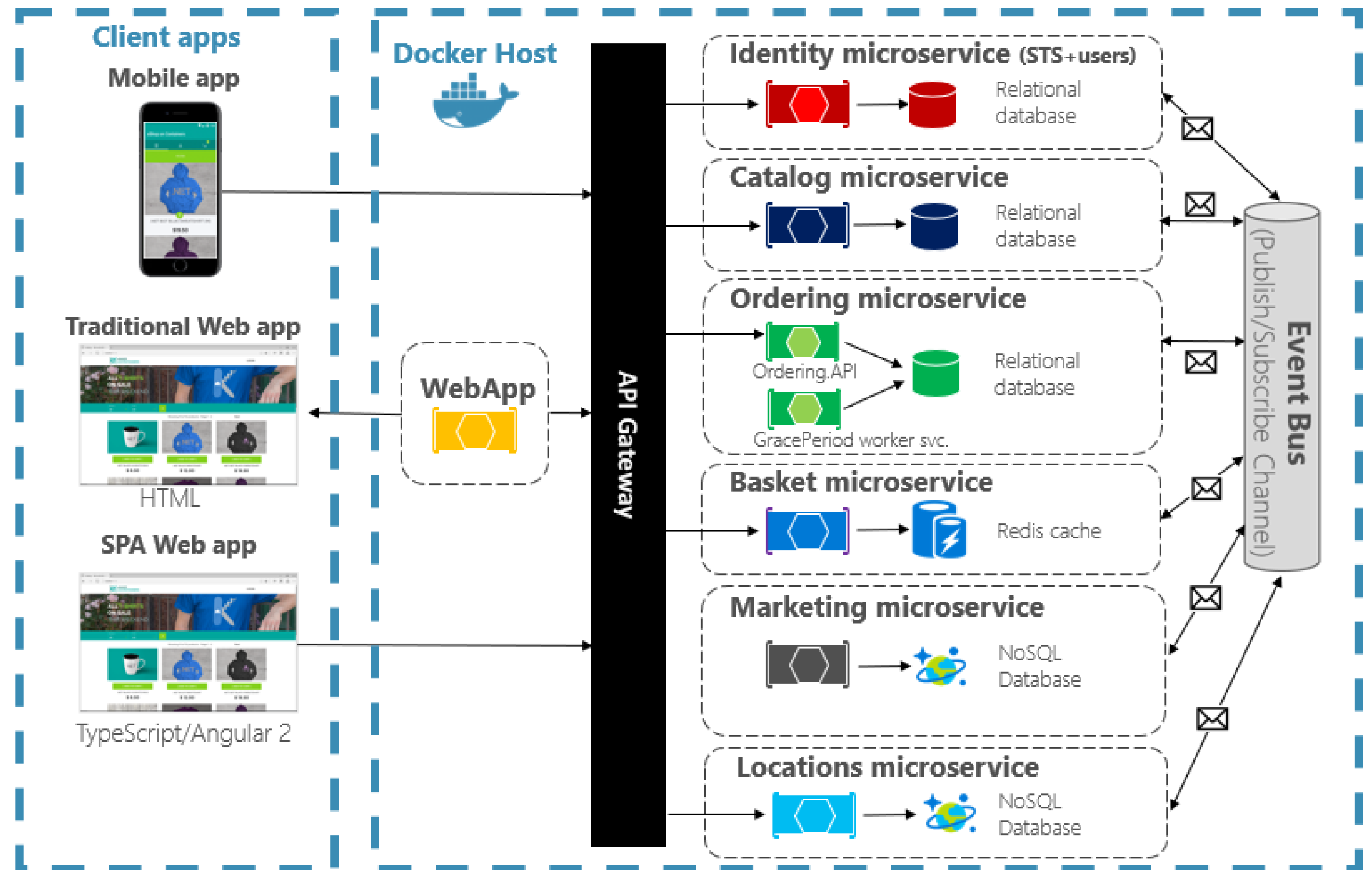
- Not commonly thought of as part of Cloud Native but they are part of the overall solution.
- Blazor
- Other SPAs
- MAUI



Cloud Native

Common

- Data Perf: EF Core 6.0 performance is now 70% faster on the industry-standard TechEmpower Fortunes benchmark, compared to 5.0.
- IO perf writing 1mb is 2x faster reading is 4x faster
- C# language features
 - Improved Lambdas, Global Usings, Top Level Statements
- Constrained Environments
- Open Telemetry
- Improved Single File
- Logging Source Generators



Container Improvements

- Honor CPU settings for Windows containers (worked on Linux before .NET 6)
- Flexibility in what Environment.ProcessorCount returns
 - Lots of algorithms in .NET tune themselves according to the value of ProcessorCount. i.e. GC, Kestrels IO threads, concurrent data structures behavior.
 - In 6 ProcessorCount returns the value of `--cpus` but you can change that to have it return a specific value to aggressively tune things.

```
C:\git\dotnet-docker\samples\dotnetapp>docker run --rm --cpus 3
42
42      ,d      ,d
42      42      42
,adPPYb,42 ,adPPYba, MM42MMM 8b,dPPYba, ,adPPYba, MM42MMM
a8" `Y42 a8" "8a 42 42P' `8a a8P_____42 42
8b 42 8b d8 42 42 42 8PP"*****" 42
"8a, ,d42 "8a, ,a8" 42, 42 42 "8b, ,aa 42,
`"8bbdP"Y8 `"YbbdP"' "Y428 42 42 `"Ybbd8"' "Y428

.NET 6.0.0-preview.3.21201.4
Microsoft Windows 10.0.19042

OSArchitecture: X64
ProcessorCount: 16
TotalAvailableMemoryBytes: 75.00 MiB
```

```
C:\git\dotnet-docker\samples\dotnetapp>docker run --rm --cpus 3
42
42      ,d      ,d
42      42      42
,adPPYb,42 ,adPPYba, MM42MMM 8b,dPPYba, ,adPPYba, MM42MMM
a8" `Y42 a8" "8a 42 42P' `8a a8P_____42 42
8b 42 8b d8 42 42 42 8PP"*****" 42
"8a, ,d42 "8a, ,a8" 42, 42 42 "8b, ,aa 42,
`"8bbdP"Y8 `"YbbdP"' "Y428 42 42 `"Ybbd8"' "Y428

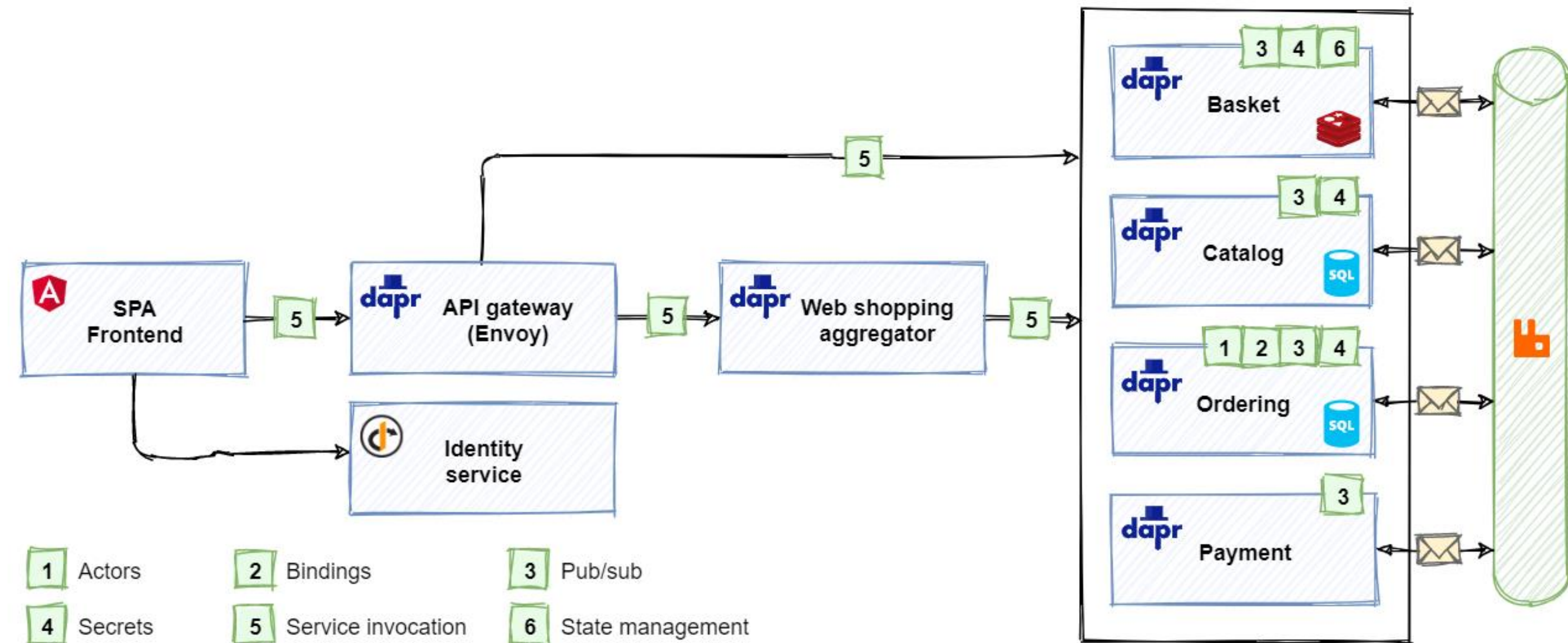
.NET 6.0.0-preview.6.21272.4
Microsoft Windows 10.0.19042

OSArchitecture: X64
ProcessorCount: 3
TotalAvailableMemoryBytes: 75.00 MiB
```

Cloud Native

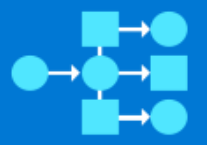





DAPR

- DAPR aims to solve common Cloud Native problems with a set of polyglot building blocks.
- <https://github.com/dotnet-architecture/eShopOnDapr>



DAPR

Use what makes sense

 Service-to-service invocation	 State management	 Publish and subscribe	 Resource bindings and triggers	 Actors	 Observability	 Secrets
Perform direct, secure, service-to-service method calls	Create long running, stateless and stateful services	Secure, scalable messaging between services	Trigger code through events from a large array of inputs Output bindings to external resources including databases and queues	Encapsulate code and data in reusable actor objects as a common microservices design pattern	See and measure the message calls across components and networked services	Securely access secrets from your application

DotNet 2021

ONLINE TECH CONFERENCE

www.dotnet2021.com

#DotNet2021

Thanks and ... See you soon!

Thanks also to the sponsors. Without whom this would not have been posible.

plain
concepts

FUNDACIÓN
GOMAESPUMA

"Educando con una sonrisa."

Microsoft

intelequia



My Public
Inbox

DevsDNA™