Case Study

Information Technology Services

MIL corp.

Integrated Applications Development Across the Enterprise

Challenge

Applications that are built in isolation create barriers in organizations to feature reuse, data integration, shared infrastructure, cost savings and development velocity. For decades, a federal government client has oscillated between developing applications in silos and targeting a single platform for all application development. Integrating data and application functionality is typically focused on creating a single source of authority for each set of data and functionality. The difficulty with this approach is all systems come to depend on the authoritative service and negotiating with every business unit to agree on a single set of standards is an infinite task. How then can we continue to develop applications that are custom tailored to specific business scenarios and domains while integrating applications and data across the enterprise?

Solution

The MIL Corporation has adopted a microservices style of architecture across multiple development teams to build applications that are specific to individual client contexts while integrating with and providing enterprise services. To support the development of microservices, MIL leverages the Microsoft Distributed Application Runtime (Dapr), Docker and the Azure Kubernetes Service, Azure Service Bus, for messaging architecture and Domain Driven Design (DDD) techniques. Applications are built by composing containerized microservice workloads that are independently developed, deployed, and operated. This allows us to seamlessly blend multiple external and internal web applications with a customer portal CMS, enterprise services, and a CRM backend without having to choose one platform to do it all. The enterprise services provide data and functionality common to the organization to all platforms and applications by connecting systems via a messaging architecture. Messaging allows each service to respond to enterprise events how they see fit and relieves services of the burden of having to be designed for more than their immediate purpose. This has kept the design simple and targeted toward delivering business requirements rapidly and iteratively over time.

Benefit

There are many benefits to developing using containerized microservices. Infrastructure costs have gone down significantly for our clients as dozens of applications are now deployed in containers in shared Kubernetes clusters. Instead of building and deploying a monolithic application or cloud-based platform to run everything, we can build services and update them independently without worrying about negative side-effects. While particular requirements and data models are developed specifically for each application, general techniques are standardized and reused from project-to-project, accelerating development. PaaS services can be plugged into the architecture to provide value-add services like single sign-on, customer logins, enterprise search, and advanced capabilities like artificial intelligence, predictive modelling, and machine learning. Common microservice chores like pub/sub, service invocation, persistence, secrets, and more can handled by the Dapr runtime so that developers can focus on building with the tools they already know. Microservices also give us a viable way to lift and shift legacy applications to the cloud or build cloud-native applications in a fully iterative way in short bursts as well as evolving them over the long-haul without destabilizing existing deployments.