Service Fabric Customer Profile: Citrix

Authored by Mauricio Caro and Nanette Ray at Microsoft in conjunction with Thomas Hammond and Bradley Rowe of Citrix

This article is part of a series (http://aka.ms/TCP) about customers who've worked closely with Microsoft on Azure Service Fabric over the last year. We look at why they chose Service Fabric, and we take a closer look at the design of their application.

In this installment, we profile Citrix Systems Inc. (https://www.citrix.com/) and their single sign-on solution built using a microservices approach on Azure Service Fabric. Also read about their business journey using Azure to implement single sign-on (https://customers.microsoft.com/story/citrix).

Based in the United States with global offices, Citrix has been at the forefront of the digital enterprise since launching their first remote access product in 1989. Today Citrix is building the workspace of the future by operationalizing the technology that businesses need. Their solutions are in use by more than 400,000 organizations including 99 percent of the Fortune 100 and 98 percent of the Fortune 500.

A Microsoft partner for more than two decades, Citrix has evolved into a leading cloud services provider. When Citrix needed help with the disparate online authentication systems used by their various cloud offerings, the Microsoft team showed them how Service Fabric and microservices could be part of the solution to streamline the customer experience with a cost-effective new architecture.

An early cloud adopter looks to streamline

Four years ago, Citrix products and services relied on deployments on premises and a packaged license model, but they knew that the cloud was the future. To evolve their business, Citrix started to expand into delivering its technology as software as a service (SaaS). For example, they migrated ShareFile to a cloud platform, creating a secure enterprise file sync and sharing (EFSS) SaaS platform. Other SaaS offerings include Podio, a team workflow solution, and Citrix Cloud, a management and integration platform.

The move to a cloud platform was a huge success for these services, giving customers greater elasticity and scale among other benefits. However, the products lacked a unified authentication mechanism. Each SaaS offering used its own authentication mechanism, identity policies, and database. Some included two-factor authentication, some didn't. Decades of product-centric business had created development silos, and customers felt the effect whenever they accessed Citrix services, many of which required different logon credentials.

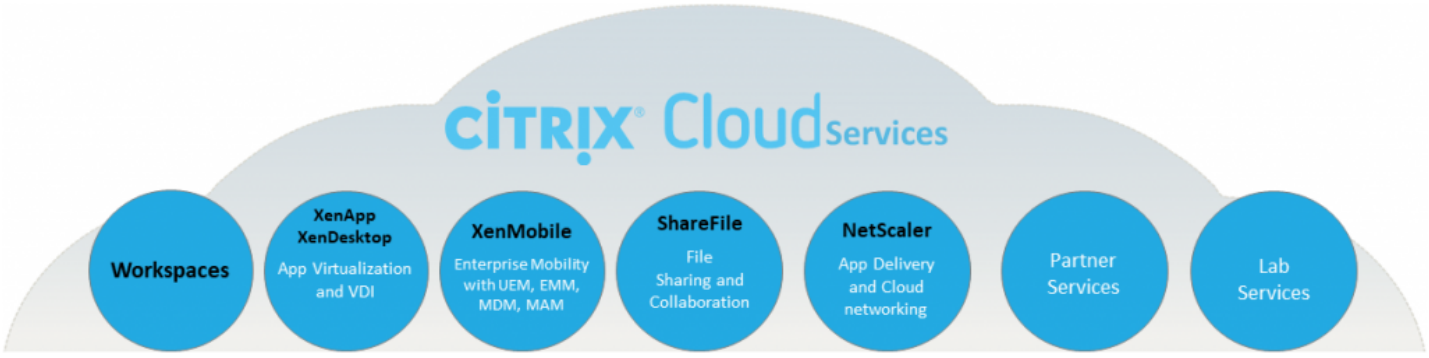

Figure 1. Citrix wanted a unified authentication mechanism for its many SaaS offerings.

Single sign-on across Citrix Cloud

With single sign-on as the goal, the Citrix Identity Platform (CIP) team got started. CIP was designed to support Citrix Cloud, a mission-critical service that handles more than 10 million Citrix users worldwide. The solution needed to unify and simplify the customer experience across the spectrum of Citrix SaaS offerings.

The CIP engineers also wanted a solution that was quick to update and deploy. Citrix Cloud is highly complex, and updates require many services to be built and deployed together in a two week release cycle. From the outset, the CIP team knew that they wanted to be able to update CIP quickly, so they needed support for a true CI/CD pipeline. That ruled out monolithic architectures.

Since CIP was a relatively small project for Citrix, it seemed like the perfect environment for a simpler, microservices-based approach. A smaller set of identity services could be deployed separately from everything else in Citrix Cloud, and each service could be built, deployed, and interacted with separately.

The plan was to build CIP as a scalable, resilient platform-as-a-service (PaaS) offering with integrated support for single sign-on services for use by other Citrix Cloud tenants. A PaaS approach would free the team from the responsibility for managing the hardware and the operating system, which is handled by the service provider (Microsoft in this case). The Citrix team wanted to focus on the authentication issue, not cloud administration.

> *"When we started the Citrix Identity Platform, we wanted to try something different and thought a pure microservice model may keep us more honest. The goal was to make the platform truly CI/CD, and that's exactly what we were able to do because of Service Fabric.*
>
> *—Thomas Hammond, Citrix Principal Software Engineer*

Service Fabric and the case for density

A microservices architecture helped the Citrix team solve another issue: density. When the CIP team began investigating PaaS hosting environments for their identity services, they first considered Azure Cloud Services (https://azure.microsoft.com/services/cloud-services/), which was already in use at Citrix. Azure Cloud Services provides cost-effective virtual machines and a complete, configured environment for deploying applications. Within that environment, developers configure the number of instances of their services they want to run, and the platform scales up and down accordingly.

The model worked well for their tiered applications, but microservices did not fit the Cloud Services model as well. The CIP team wanted the flexibility to pack more microservices on fewer nodes. Densely packed services would provide the scale they needed to support millions of sign-ons without paying to scale out on so many individual virtual machines. If services are tightly bound to the host in a 1:1 relationship, much of the flexibility of a microservices-based architecture is lost.

Although Azure Cloud Services provided part of the solution, Service Fabric turned out to provide the complete environment that Citrix was looking for. Service Fabric is designed with distributed architectures in mind, and enabled Citrix to create 100 services yet run them on only 10 nodes. The CIP team could run a cluster of any size they liked and keep costs in check.

Citrix engineers were in San Francisco in 2016 for the //Build Developer Conference, where Microsoft announced that Service Fabric was generally available as a service in Azure. Service Fabric had been in production use at Microsoft for five years, but now it was publicly available as a mature microservices application platform. With support for lifecycle management, stateless and stateful services, density, and performance at scale, Service Fabric had everything the CIP engineers were looking for.

Citrix immediately started using Service Fabric for CIP and began working it into with their existing testing and development processes, eventually building their integration systems around it.

Find out more about the differences between Cloud Services and Service Fabric (https://docs.microsoft.com/azure/service-fabric/service-fabric-cloud-services-migration-differences).

CIP architecture based on Service Fabric

CIP is an implementation of OpenID Connect, a simple identity layer built on top of the OAuth 2.0 protocol. It authenticates users in different ways, generates access tokens for use in various systems, and provides a single-sign-on solution for Citrix products.

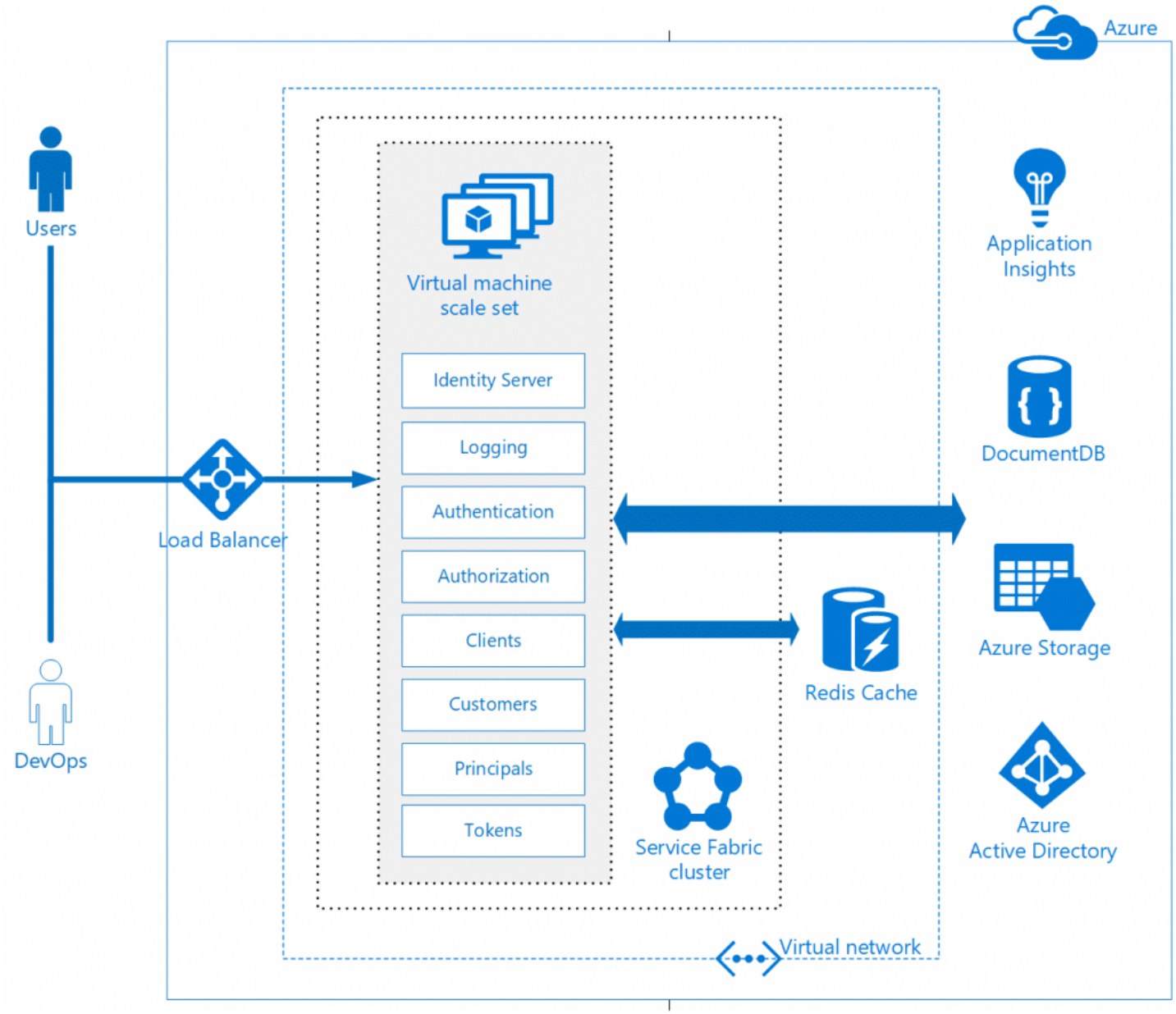

Figure 2. Leveraging Service Fabric to build the Citrix Identity Platform (CIP) on Azure.

An earlier implementation of the architecture included two virtual machine scale sets: one for the front end and one for the back end. The team thought they would need the additional scale to support the load. However, they discovered it was simpler—and just as scalable—to use only one scale set. The simpler deployment was easier to manage, and better still, running the instances of their microservices in a single scale set enables the team to deploy identical environments for development, testing, integration, and staging.

After the move to a single scale set, the team kept a close watch on performance but didn't see any negative effects. They can quickly scale up the entire platform by adding virtual machines and get as much performance as they want. Service Fabric balances the load across nodes perfectly—a huge win.

Mission-critical microservices

CIP is designed to handle identity management and authorization through integration with Azure Active Directory. CIP uses microservices to accept credentials, create principals, and return signed tokens. In addition, it can federate authentication to other directory services.

The microservices were designed to encourage internal business units at Citrix to create their own plug-ins and enable federated authorization to their products. The product teams support various types of authorization, so flexibility was important, as was isolation. The Service Fabric clusters are isolated by a virtual network that is separate from other services. The product teams can experiment with CIP and write microservices that run in an isolated service without affecting CIP. Service Fabric makes implementation of these product-specific services a straightforward, controlled process for the CIP engineers.

Identity server

This service takes input from a customer logon and sends it to the back-end Service Fabric instance for authentication and authorization. The identity server service includes a REST endpoint and serves the OAuth endpoints used by the authentication services. In that sense, the service works like an O-end web app with an SSL endpoint on port 443. Relying parties use that port to connect to the back-end services, which are isolated behind a virtual network.

Authentication and authorization

CIP is an OAuth provider and works with the management systems used by Citrix products. The Authentication microservice is a brokered service based on security token service and claims-based authentication. The authorization microservice verifies that a client has permission to perform a certain action.

Clients and customers

The clients service verifies an open authentication (OATH) client and shows the platform that a relying party can make use of CIP services. This service also maintains the client data.

In Citrix Cloud, a customer is a tenant. The customer microservice's main purpose is to unify customers across Citrix services. The microservice aggregates the tenant identity data provided by Citrix Cloud products, then stores the data in one place. The customer microservice enables CIP to become the central nexus of identity at Citrix, hosted entirely by Service Fabric.

Principals

The core of CIP is the principals microservice, which stores individual user information as compared to tenant information stored by the customers microservice. User credentials are verified against the principal service to determine who's who, because people use multiple ways of authenticating. For example, Citrix users may have one ID for NetScaler, and a different one for XenApp. The CIP engineers developed a motto: "Single principal, multiple identities." The principal service was designed to handle these cases.

The principals server also helps provide integration with other products. Citrix Cloud checks user identity against the principals service. Now, when a NetScaler or XenApp user signs on to another service such as ShareFile, his user credentials get stored under the principal service and his identity goes with him in Citrix Cloud.

Benefits of Service Fabric

As the host for CIP's microservices framework, Service Fabric provided many advantages for the Citrix engineers, enabling them to:

- Build once and reuse many times with different use cases and applications.

- Support autonomy so teams can make changes faster.
- Build products faster with more flexibly through composition.

In particular, Service Fabric supported the overarching goals for the new identity platform:

- **Federated identity for single sign-on:** CIP secures the Service Fabric cluster through Azure Active Directory, which supports the directory federation needed for single sign-on. The Citrix engineers wanted to base their authentication system on the familiar Active Directory services used by many of their existing customers, which would lower the bar to entry. The strategy worked, and a growing number of product teams are now integrating with CIP to provide single sign-on for Citrix Cloud users.
- **Density:** CIP uses Service Fabric and virtual machine scale sets to scale out, allowing the engineers to proactively manage their Service Fabric cluster without impacting the bottom line. Scale sets are sets of virtual machines that are configured identically and scale in and out rapidly and automatically. Citrix no longer needs to provision virtual machines separately. Only the compute resources needed by CIP at any time are used, a great cost benefit. As services are added and the density increases, they can dedicate new servers that spread the work evenly in a way that's transparent to the rest of the environment.
- **Self-healing:** When failures occur, service recovery is critical so customers can continue their work. Service Fabric can self-heal by moving services to healthy nodes, ensuring persistent data is replicated and transparently orchestrating the process. This feature helped the team meet their recovery time objective of under 5 minutes.
- **Upgrade management:** Multiple teams at Citrix develop and deploy new services at different times, so the single sign-on platform needed to support an upgrade process that wouldn't impact these teams or their products. Service Fabric's incremental upgrades helped simplify the release management workflow.
- **Performance:** The overall CIP solution was optimized for its primary task, to authenticate users, but it had to meet the uptime service levels guaranteed by Citrix Cloud (99.95 percent). Service Fabric's performance is based on the underlying Azure virtual machines and storage resources, which helped CIP meet the SLA.

> *With our single sign-on solution, if a Citrix customer already has Azure Active Directory, they can now just configure federation and that's it. Customers can be easily onboarded with an authentication schema that they know.*
>
> *—Bradley Rowe, Citrix Software Engineer*

A DevOps approach

By the time the CIP project started, Citrix had already reorganized the engineering teams around the DevOps model. One of the original project goals—to adopt microservices—was meant to improve their agility. Smaller pieces of code can be divided among teams in support of quick release cycles.

The move to DevOps was a four year journey for Citrix and today is a core practice of their software development process. When the engineers first adopted Service Fabric, they incorporated it into the existing CI/CD process, which at that time consisted of an integration environment, testing, and a handful of services. As they took advantage of more Service Fabric features, their processes began to change for the better.

Today they have twice as many services but greater reliability. Services continue to be added as more business units adopt CIP, and every change gets tested against components in the pipeline. DevOps practices enable engineers to create an environment that allows all teams to drive towards repeatable processes.

In their journey to DevOps, Citrix ultimately eliminated testing as a separate role. All software engineers develop and test, maintaining responsibility for their code all the way to production. The next improvement they plan is a fully automated integration, staging, and deployment environment.

> *Our new development platform maps perfectly to the DevOps model. Service Fabric has no competition in that sense.*
>
> *—Thomas Hammond, Citrix Principal Software Engineer*

Summary

Citrix had great success in the cloud with multiple products and services used by millions of users worldwide. But customers wanted the ease of a single-sign-on solution. Using Service Fabric as the basis for a new, microservices-based architecture, Citrix create a single sign-on authentication platform for Citrix Cloud that is being adopted by a growing number of their product teams.