

ZEISS Group focuses on Azure Service Fabric for key integration platform

January 25, 2019



The specialists who develop and manufacture the digital instruments at ZEISS Group have plenty on their minds. The sophisticated instruments they build need to work reliably and stay up to date. To serve business customers and end users better, the ZEISS Group built an Azure-based digital integration platform with [Azure Service Fabric](https://docs.microsoft.com/en-us/azure/service-fabric/) (<https://docs.microsoft.com/en-us/azure/service-fabric/>) at its center.

"Some of our devices such as the robotics instruments used in microsurgery are highly complex. We didn't want to give the engineers developing these devices and other customer touch points the added headache of integration. Service Fabric was a key piece in making our digital integration platform as simple as possible."

—Kai Walter: Lead Solution Architect

Carl Zeiss AG

Headed by Carl Zeiss AG, a company with a venerable history that includes the invention of the compound microscope in 1857, the ZEISS Group performs extensive research and development in optics and photonics. As pioneers in scientific optics, ZEISS develops, produces, and distributes measuring technology, medical technology, eyeglass lenses, camera and cinema lenses, binoculars, and semiconductor manufacturing equipment.

To streamline the customer experience, a development team at the ZEISS Group embarked on a digital integration project. The goal was to build an interoperability platform that allows field devices to connect to back-end business systems from SAP, Salesforce, and others, where customer records, device data, service history, and other information was stored. Customers would then have access to the latest software updates and support information.

The solution needed to be simple. The ZEISS team did not want to add to the complexity faced by the users of their highly advanced devices. Yet the small development team wanted to work quickly, so they sought help from PlanB. GmbH, a Microsoft Gold partner that specializes in cloud infrastructures and Azure implementations. PlanB. helped the team to create an Enterprise Service Bus in the cloud, capable of integrating the various devices and customer touch points via standard REST-based APIs. Reusable APIs could also be used by other developers and partners worldwide to connect devices to the customer service, shopping, and knowledge assets stored in the back-office systems.

For more information about the business drivers behind this platform and its user credential service, see the [Zeiss business story](https://customers.microsoft.com/story/zeiss-manufacturing-azure) (<https://customers.microsoft.com/story/zeiss-manufacturing-azure>) .

A layered Azure sandwich

The digital integration platform was designed to connect devices and other customer touch points (the consumers) in the simplest way possible given the various back-end systems (the producers). The solution combines several Azure PaaS (platform as a service) offerings. If the Azure Marketplace had a useful service, the team was willing to try it. This experimental approach provided a working prototype quickly while minimizing the typical costs and time associated with spinning up traditional servers and services.

In addition, the team was still learning about what Azure could do. Azure PaaS gave the team the perfect low-risk, low-cost laboratory to learn in.

Through this process of explore-expand-extract, the ZEISS team discovered Service Fabric, the environment for building large-scale, highly available, distributed cloud applications. Service Fabric works like an

orchestration engine for deploying microservices across a cluster of machines and provides developers with the tools they need to design apps and services that run at cloud scale.

After a few weeks of continuous evolution, the developers came up with the initial design, which placed Azure services in the middle of a sandwich with APIs on top, and back-office systems on the bottom. At the top layer, consumers interact with the system through the APIs that are exposed through Azure API Management. This layer plays a key role in hiding the complexity of downstream transactions from consumers, and makes it easy for the ZEISS team to manage the API life cycle. As new consumers are added to the platform, the ZEISS development team works with them to create the APIs they need. With the capability to mock up responses based on the joint API agreement, this layer decouples the development teams, enabling consumer teams to begin working with the APIs immediately regardless of back-end development.

The middle layer transports, transforms, and holds state for the information passing through the digital integration platform. For stateless scenarios, API Management passes incoming messages to Azure Functions, which performs the transformations needed to enable consumers and producers to communicate—for example, transforming from queues to REST APIs and vice versa.

The development team chose Functions because of its simple programming model with declarative bindings and few lines of code. They could focus on writing the transformations they needed with less risk and grow with the innovations in the Azure Functions host environment. and allow other services to handle infrastructure complexity. However, Functions, when run in a consumption plan, does not support virtual network integration. So they created an Azure DevOps pipeline to package the function code together with the Functions runtime into containers and to handle deployment. The developers didn't have to change any of their code.

Service Fabric handles their stateful services and provides a runtime environment that supports their virtual network. To hold state for long-running, multiple step processes, the developers used an actor-pattern-based application framework built on top of Service Fabric Reliable Services. Each Reliable Actor service is a partitioned, stateful Reliable Service. Information is then communicated to the appropriate back-end system, which matches it to the desired result and reports back via the API to Service Fabric, which communicates the result to the consumer. This layer also includes Azure Service Bus queues to support asynchronous messaging between services that require it.

In addition, some information is persisted via Azure Cosmos DB, a multi-model database service on a globally distributed, managed platform. When Azure Cosmos DB is used in combination with the binding capabilities in Azure Functions, developers can store and retrieve data with much less code than is required for storing data in a traditional relational database.

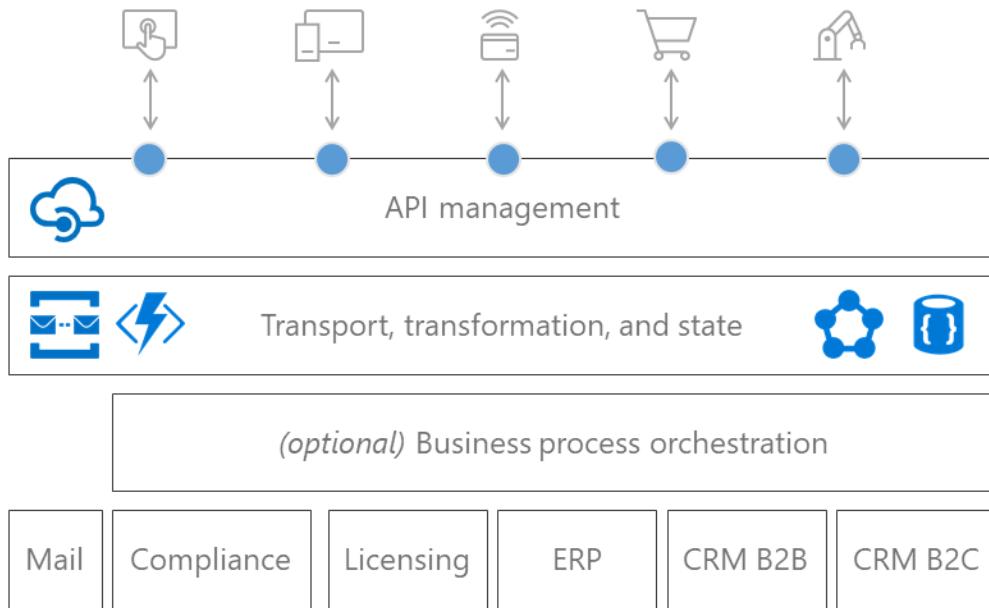
For more complex processes, a business process layer based on SAP services orchestrates data and processes for enterprise resource planning (ERP) in the downstream systems.

The bottom layer includes the back-end producers such as the SAP and Salesforce enterprise systems.

" Azure allowed us to grow. We started with public services then moved into private virtual networks, step by step."

—Kai Walter: Lead Solution Architect

Carl Zeiss AG



Evolution of the digital integration platform

At first, the developers used Service Fabric only for the use cases where they needed to store state and retain it for the duration of a transaction. Service Fabric makes state highly available without the need to persist state to an external store. As the team became more familiar with Service Fabric, they began relying on it in more use cases.

Stateless services were written as small pieces of code in Azure Functions—another key PaaS component. Azure Functions is a serverless compute service that enables developers to run code on demand without having to explicitly provision or manage infrastructure. The ZEISS developers wanted to spend their time perfecting the services of their integration platform, not thinking about the infrastructure. With this in mind, they are evaluating [Service Fabric Mesh](https://docs.microsoft.com/en-us/azure/service-fabric-mesh/service-fabric-mesh-overview) (<https://docs.microsoft.com/en-us/azure/service-fabric-mesh/service-fabric-mesh-overview>) , a fully managed service that frees developers from cluster management.

Azure Functions ran in a consumption plan that automatically allocated compute power when the code was running. It scaled out as necessary to handle the load, and then scaled down when the code was not running. However, the team wanted more control over scaling. After six months, they switched to self-hosting of the Azure Functions runtime—a move that also gave Azure Functions the ability to directly access the back-end services running in the ZEISS datacenter.

As a security precaution, transactions involving certain types of data do not leave the corporate network. When the team set up Service Fabric, they also created an ExpressRoute connection from the Azure virtual network to the ZEISS datacenter. This connection enabled Service Fabric to operate as if it was running in the datacenter. Further, it gave the developers access to datacenter resources and added control over where transactions occur, such as those handled by Service Bus and Azure Cosmos DB.

The developers wanted to give Azure Functions greater access to the back-end systems but keep the lambda notation and declarative development paradigm of Functions. They realized they could host Functions in containers within Service Fabric, which was already connected to the ZEISS datacenter. This move made Service Fabric central to all integration tasks, not just the stateful transactions.

Services in the digital integration platform

The success of the integration platform was driven in large part by three critical stateful services that run in Service Fabric:

- **Software Update service.** This stateful service processes customer requests to upgrade their software, a feature that ZEISS offers on demand for their digital instruments. This service triggers and waits for order and license processing on several back-end systems, issues a license file directly to the requesting customer, and collects and preserves the status for better traceability.
- **ZEISS Global Mail API Service.** The platform enables cloud-based consumers to send authentic “ZEISS” email over the corporate IT email back-end. These consumers include web shops and devices that would

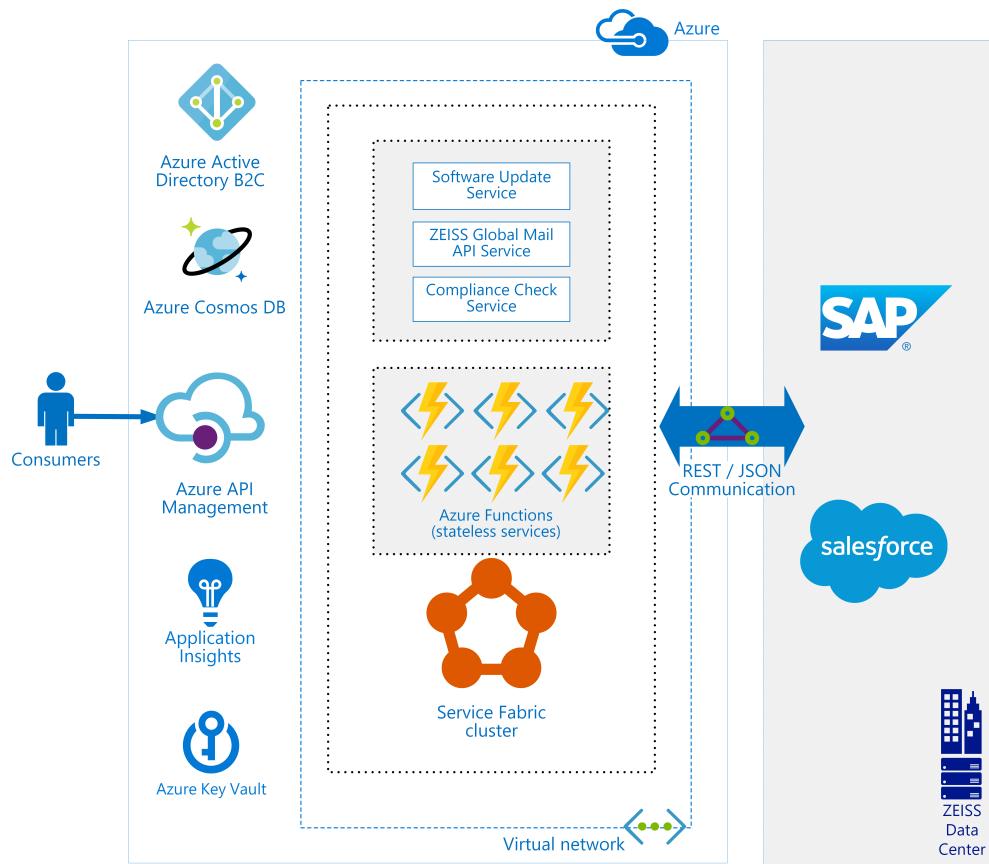
not otherwise have direct access to the company's back-end email service. This stateful service waits for completion of the email sending process and can notify the original requestor upon success or failure. Designed as a simple-to-use entry-level service, it also entices more developers to sign up for and use the integration platform APIs in their work. From the ZEISS developer portal, users can immediately sample the service, send a test email message using the programming language of their choice, and see for themselves how easy the APIs in general are to use.

- **Compliance Check service.** This stateful service runs a compliance check for each transaction involving delivery of goods that may be subject to export restrictions. The service verifies whether the goods can be delivered to a particular address, country, or company.

"Our intention was to start fast and start small, and so we used Azure PaaS first. When we hit certain limitations of PaaS, we discovered Service Fabric. It became the centerpiece of our more sophisticated scenarios."

—Kai Walter: Lead Solution Architect

Carl Zeiss AG



Many Azure services make a sandwich

ZEISS is heavily invested in SAP enterprise systems, and the digital integration platform needed to integrate with several, including SAP HCM, SAP ECC, and SAP CRM. ZEISS passes about a half-million SAP messages a day through the platform. As a result, balancing load proved to be a critical feature, because the enterprise systems on the back end could not operate at cloud speed, and some of the consumers couldn't handle the number of messages sent by SAP either.

The platform adjusts automatically to the load by using different messaging patterns to enable communication across the layers:

- **Synchronous:** A consumer can send a request through API Management, and the platform routes it to producer, performing basic transformations and applying authentication or token validation as needed.
- **Asynchronous in:** When producers cannot handle the rate of incoming traffic, a request is placed in a Service Bus queue. A function reads the queue and communicates at a pace the producer can accept, sending only one message at a time from the queue.
- **Asynchronous out:** A producer can send a message via API through a function back to a queue for asynchronous processing. The function then pushes it to the consumer.
- **Decoupled:** Some consumers cannot accommodate the number of documents sent by the SAP producers. In these cases, the documents are buffered in Azure Cosmos DB until the consumer pulls them using an API call.

To keep upstream and downstream traffic running smoothly, Service Fabric is at the center of all transactions. When an SAP transaction occurs, the data must be unloaded as fast as possible to avoid placing undue load on the servers. The upstream flow of data must be handled in the most performant way possible. So, the ZEISS developers used the sandwich of Azure components to serve as a buffer. Users and devices communicate through the API, where Azure API Management throttles and measures the API calls to the platform.

Other Azure components are used to throttle downstream. On the transport layer, Azure Cosmos DB is used to persist state or buffer documents passing through the platform from some SAP producers back to consumers.

In addition, the digital integration platform used a few other Azure services:

- [Azure Active Directory B2C](https://docs.microsoft.com/azure/active-directory-b2c/active-directory-b2c-overview) (<https://docs.microsoft.com/azure/active-directory-b2c/active-directory-b2c-overview>) is the identity management service used to authenticate ZEISS customers on apps, portals, and shops. It works in conjunction with token validation performed on API call-level safeguards to provide authorization to use certain back-end capabilities.
- [Active Directory Federation Services](https://docs.microsoft.com/windows-server/identity/active-directory-federation-services) (<https://docs.microsoft.com/windows-server/identity/active-directory-federation-services>) is the identity management service used to authenticate ZEISS employees who log on to the developer portal for the digital integration platform.
- [Azure Key Vault](https://docs.microsoft.com/azure/key-vault/key-vault-overview) (<https://docs.microsoft.com/azure/key-vault/key-vault-overview>) stores cryptographic keys and secrets associated with various platform services.
- [Azure Application Insight](https://docs.microsoft.com/azure/application-insights/app-insights-overview) (<https://docs.microsoft.com/azure/application-insights/app-insights-overview>) (<https://docs.microsoft.com/azure/application-insights/app-insights-overview>) enables the team to monitor the platform, diagnose issues, and keep track of API usage and life cycle.
- [Azure DevOps](https://azure.microsoft.com/overview/devops/) (<https://azure.microsoft.com/overview/devops/>) and the entire Azure service ecosystem support efficient development based on an infrastructure-as-code approach using scripts and templates.

Key advantage in Service Fabric

Service Fabric is the center of the ZEISS digital integration platform and offered the development team several key benefits:

- **One compute platform.** Service Fabric runs all transactions at the scale they require.
- **Reliable Actors.** An application framework based on the Reliable Actor service holds state in a consumer context, which was necessary for transactions that take place across several steps in the Compliance Check and License Update services.
- **App management.** The developers integrated Service Fabric with the continuous integration and continuous deployment (CI/CD) capabilities of Azure DevOps. Service Fabric provides support for the full application life cycle and CI/CD of cloud applications.
- **A platform that grows with the team.** The ZEISS development team turned to Service Fabric use case by use case. Within six months, using an evolutionary approach with little capital binding and low capital risk, the team was able to deliver a fully fledged integration platform.
- **Deploy—and redeploy.** The team learned by trial, measuring and improving each iteration and making the most of Azure PaaS and the ability to turn services on and off. With Service Fabric, the developers can delete deployed services as easily as they can provision, deploy, monitor, and patch them.

- **Cost-effective performance.** In moving Azure Functions to Service Fabric containers, the team got an added benefit. They pay once to process both stateless functions and stateful Service Fabric programming models.

Conclusion

Azure services, and Service Fabric in particular, are at the center of a digital transformation at ZEISS. The digital integration platform supports a visionary effort to make smart devices and prepare ZEISS for the future. With secure access to ZEISS enterprise systems, consumers in the digital integration platform can schedule their own service calls and give ZEISS designers predictive insight into customer needs.

With the digital integration platform in place, now the team at ZEISS is investing in developing standard patterns and building integrations in a factory approach.

“Service Fabric was the key because it allowed us to run all our stateful and stateless interactions on one compute platform that we can scale on our terms. Everything is in Service Fabric.”

—Kai Walter: Lead Solution Architect

Carl Zeiss AG