

Societe Generale's complex financial simulation platform expands on Azure Service Fabric architecture

April 20, 2018



Authored by Stephane Bonniez from Societe Generale; Gregoire Sirou, Nicolas Duran, and Erik Ferrand from Qarnot Computing; in conjunction with Eric Grenon from Microsoft.

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

In this installment, we profile [Societe Generale \(https://www.societegenerale.com/en/home\)](https://www.societegenerale.com/en/home) and [Qarnot Computing \(http://www.qarnot.com/\)](http://www.qarnot.com/), their grid computing application, and how they designed the architecture.

Societe Generale provides financial services to 31 million individuals and professionals worldwide, placing innovation and digital technology at the heart of its activities. Its corporate and investment banking business, SG CIB, offers global access to markets through solutions for equities, fixed income and currencies, commodities, and alternative investments. Their global markets platform is recognized for its worldwide leadership in equity derivatives, structured products, euro fixed income markets, and cross-asset solutions.

Societe Generale partnered with Qarnot Computing, and the Microsoft Azure team to build a new financial simulation platform. Market activities require complex financial simulations that run on large-scale grid computing infrastructures. The new platform is flexible, scalable, environmentally responsible, and designed to support the growth of Societe Generale's business in a rapidly changing economy.

Founded in Paris in 2010, Qarnot Computing is a pioneer in distributed cloud and smart-building technologies. They invented an innovative computing heater, the first of its kind, that uses the heat generated by the CPUs to heat buildings for free. Since 2014, more than 100 French homes, schools, hotels, and offices are heated with Qarnot Q.rads heaters. Their ingenuity has garnered several awards, including the 2015 Cloud Innovation World Cup Award.

Qarnot provides cloud computing through a distributed infrastructure where computing power is no longer deployed in concentrated datacenters but spread throughout the city in the form of heaters and boilers. Their remote cloud computing powers private and public companies, including major banks, 3D animation studios, and research labs. But when Societe Generale contacted Qarnot with their game-changing request for more compute power, Qarnot needed help from another cloud provider.

A financial simulation platform

Financial simulations are computationally intensive. They typically involve several thousand calculation tasks, taking from a few seconds to several minutes each to compute. They can also require hundreds of megabytes of data such as the historical values of equity shares over several years. But each task usually uses only a small portion of that data.

Simulation jobs are triggered by users at any time during working hours. Since Societe Generale has offices all around the world, that means at any time during the day, any day. Some of the simulations also have strong computation time constraints.

Societe Generale and Qarnot designed a solution that:

- Exposes a simple REST API to client applications within Societe Generale.
- Handles calculation jobs ranging from a few tasks to several thousands (from seconds to hours).
- Provides caching of financial data for efficient dispatching of tasks.
- Scales with the number of jobs and tasks.
- Is available around the clock.

These achievements take place in a context where new software is delivered frequently, because simulation libraries evolve continuously. Service Fabric provides a store to manage versioning and serve as repository of all binaries for the microservices and related configuration files. In addition, infrastructure costs must be kept as low as possible, although thousands of CPUs may be required to perform some simulations.

To meet these requirements, the new platform provides the following key components:

- A HTTPS web gateway exposing simulation services as a REST API.
- A collection of microservices handling data caching and the orchestration of simulation jobs, from the dispatching of tasks to the retrieval of the results.
- Several grid computing providers. Currently, [Azure Batch](https://azure.microsoft.com/en-us/services/batch/) (<https://azure.microsoft.com/en-us/services/batch/>) and Qarnot Computing's platform are targeted, but new providers can be added very easily, and internal dispatching guarantees that a job will always find room to run at the best possible price.

The web gateway and the microservices are native Service Fabric applications, all deployed in a scalable cluster in the Azure cloud.

“ With Service Fabric, we were able to build a robust, stateful microservice architecture in no time, giving us more time to focus our efforts on our product. ”

—Nicolas Duran: Senior Software Engineer
Qarnot Computing

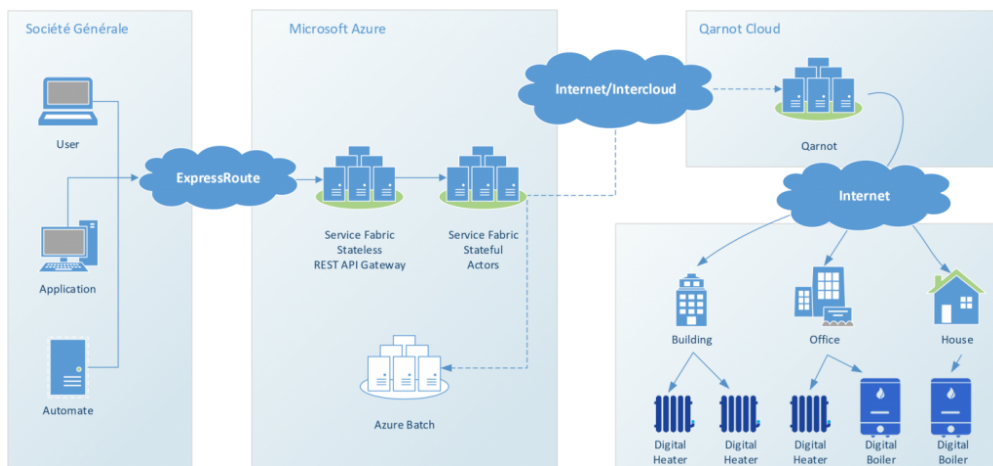


Figure 1. High-performance financial calculations are broken into discrete jobs and tasks by microservices running on Service Fabric, then distributed to available cloud computing environments.

Service Fabric implementation

The Service Fabric part of the application is written in C#, with mix of services and actors, both stateless and stateful.

The web gateway is a [stateless reliable service](https://docs.microsoft.com/en-us/azure/service-fabric/service-fabric-reliable-services-introduction) (<https://docs.microsoft.com/en-us/azure/service-fabric/service-fabric-reliable-services-introduction>). As the unique entry point of the application, the service must be highly scalable so multiple client applications within Societe Generale can run simulations concurrently. When the load increases, it's simple to add new nodes to the cluster, and Service Fabric automatically launches more gateways and balances the load across the cluster.

Calculation jobs and tasks are implemented with [stateful reliable actors](https://docs.microsoft.com/en-us/azure/service-fabric/service-fabric-reliable-actors-introduction) (<https://docs.microsoft.com/en-us/azure/service-fabric/service-fabric-reliable-actors-introduction>). For instance, each task that is dispatched to the Azure Batch or Qarnot Computing platforms is materialized as an actor. Actors are easy to write, and they have several useful properties:

- Their state is replicated on several instances across the cluster, so they are reliable, highly available, and persistent.
- They are automatically distributed across the Service Fabric cluster, which provides scalability and load balancing.
- If they have been inactive for some time, actors are automatically unloaded from memory to disk, then automatically rehydrated in memory when called again. This feature saves memory and helps scale to more actors (so more simulation jobs and tasks are supported).
- Their threading model guarantees that their state will always be consistent.

“ With Service Fabric, developers can focus on business needs and rely on the platform for resiliency, load balancing, and scalability. We can deliver better software, and do it faster.”

—Stéphane Bonniez: Project Manager

Societe Generale

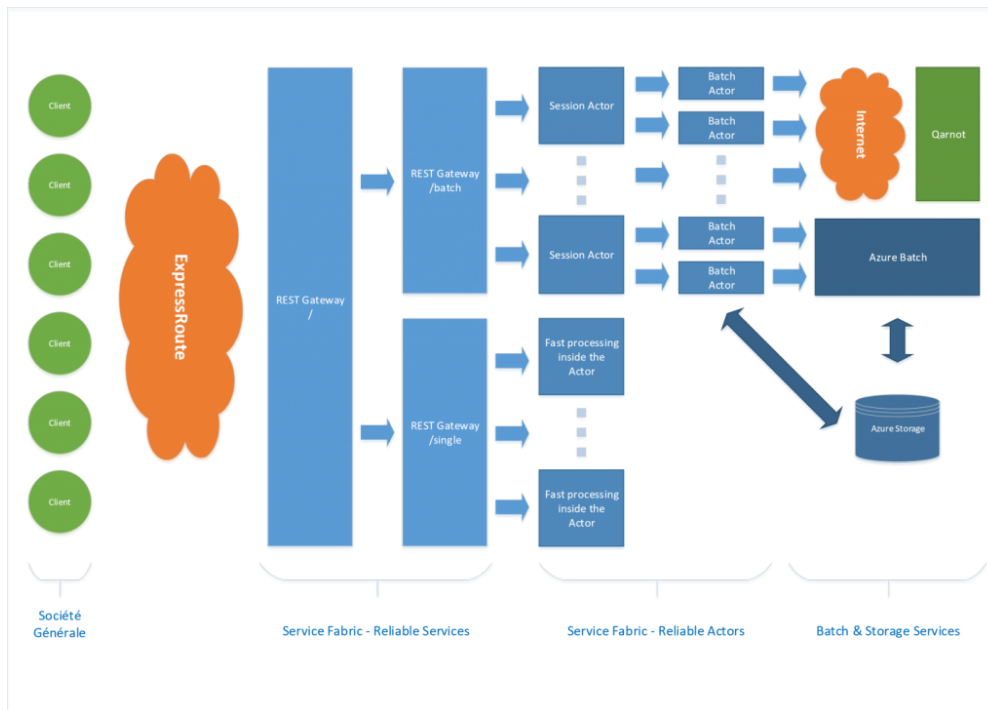


Figure 2. The solution uses both the Service Fabric Reliable Services and Reliable Actors frameworks.

Advantages of Service Fabric

With a tight schedule, the joint Societe Generale and Qarnot team needed to ramp up fast. Service Fabric offered a complete toolset with its sophisticated runtime for building distributed microservices and its complete application management package for provisioning, deploying, monitoring, upgrading, and deleting deployed applications.

The fully platform as a service (PaaS) cluster in the Azure cloud leaves the deployment and patching burden of the underlying software to Azure.

Given the deadline, the following Service Fabric benefits proved especially helpful:

- **Speed of development:** The powerful programming models provided by Service Fabric made it very easy for the developers to concentrate on business logic. Service Fabric managed the critical technical details- -replication, resiliency, deployment systems, and more.
- **Self-healing:** The calculation solution required high resilience and availability. Service Fabric's ability to provide self-healing was a big benefit. For example, if a node or a process fails, the system automatically

starts new instance.

- **Reliability:** Financial simulations involve many calculation tasks that depend on the same data, so an easy way to optimize the application was to hold a copy of this data. A client application can send all the data it will need, then the tasks it wants to compute. A cache like this wouldn't be of much use if it had to be rebuilt each time a node in the cloud is lost. Fortunately, Service Fabric makes it easy to write and manage reliable services. The developers used the Reliable Collections to handle data replication, so application code doesn't have to deal with data management. Developers simply specify how many times to replicate a state across nodes for reliability. In case of failure, Service Fabric automatically fails over to another consistent replica, and the calculation does not lose progress. This enables the client to avoid restarting the whole financial calculation.
- **Programming model:** Societe Generale and Qarnot took advantage of the productive programming models in Service Fabric to develop key components of their solution, from the gateway stateless service to the stateful calculation task service to the reliable actors used for task distribution.
- **Scalability:** Service Fabric provided the scale needed for the calculations, from one actor to thousands of actors. The developers saved countless hours--there was no need to manage the scale at the application level.
- **Application lifecycle:** The team can easily deploy a new version of the application with no downtime or deploy multiple instances of the same application. The flexibility of the cloud and of Service Fabric development tools allowed the team to fully integrate build, test, and deployment into Societe Generale's continuous integration pipeline. Code is built and packaged, then tested on a local Service Fabric cluster. If all goes well, it is automatically deployed to a development cluster in Azure. The same tests can run against the local cluster and the development cluster in Azure, which allows the team to spot bugs very early in the chain. When a version has been validated, it can be deployed to the production cluster the same way it was deployed to the development cluster, and the same tests can be used to check that deployment went well.

“With Service Fabric, Societe Generale and Qarnot were able to speed up debugging and scaling, thanks to the on-premises deployment and the perfect integration with the development tools.”

—Grégoire Sirou: CTO

Qarnot Computing

Summary

The challenge for Qarnot Computing and Societe Generale was to deliver a secure, modular, scalable, and resilient application in a very short timeframe.

Service Fabric was the right choice for the job. Its powerful toolbox handled the mechanics, so the development team could concentrate on business logic. The result is an innovative and high-quality solution that required just the right amount of effort to develop.

Now that the simulation platform is in production, the team can focus on integrating new types of simulations and scaling the platform to handle them. The goal is to enable more client applications to move away from legacy systems. The current platform lets clients use the computational capacities they want. The next stage is centered on client management. Future versions will integrate per-client capacity management and billing.