GE Global Research

GE MIGRATION TO A MICROSERVICES PLATFORM FOR MANUFACTURING AND ANALYTICS

THOMAS D. CITRINITI

APRIL 2015

Public (Class 1)

Technical Information Series

GE Global Research

Technical Report Abstract Page

Title	GE Migration to a Microservices Platform for Manufacturing Analytics		
Author(s)	Thomas D. Citriniti	**Phone**	
Component	Software Sciences & Analytics, Niskayuna		
Report Number		**Date**	April 2014
Number of Pages		**Class**	Public (Class 1)
Key Words	brilliant factory, digital thread, micro services, architecture, SOA, big data, analytics, predictive		

Abstract:

The SEI SATURN conference is having a pre-conference workshop on Microservices architecture to explore the many aspects that are emerging with this new technology. This paper will introduce the current architecture and requirements of GE with respect to their Brilliant Factory[1] and Industrial Internet[2] initiatives.

GE is a business of businesses, each having its own needs and requirements for visibility into their internal systems and ability to schedule and optimize their operations for maximum efficiency. The need to have access to the current increased volume of collected data means a change in data storage and analytics that are prevalent in current enterprise architectures. Enterprise platforms that both collect and process data will soon be overwhelmed with requests and collected data. A more distributed and less centrally based approach to overall business analytic processes is needed.

Microservices are proposed as a solution to some of these concerns and risks of the new Industrial Internet age. This paper will explore some of GE's concerns with respect to current needs and trends and how a change in architecture and low-level functionality is needed to address long-term goals. The workshop at the SEI SATURN conference will attempt to discuss and understand how Microservices may be the basic building pattern to address long term business needs.

---

[1] http://www.gereports.com/post/77834521966/meet-your-maker
[2] http://www.ge.com/stories/industrial-internet

Description:

GE is largely a manufacturing business building large complex assemblies of systems to solve large scale problems. Today, the practice for designing a manufacturing process and system for new products is mostly based on historical practices. This is largely due to the complexities of the design process. Current manufacturing systems and design processes are mostly located in very disparate locations and utilize many data systems. Operations are based on existing technologies that tend to concentrate all functionality in data centers.

The industry trend in software systems that are utilized as controls and analysis of these machines have been mostly developed in separate business groups and system divisions. This leads to a segmentation of features, service definitions, and storage mechanisms. The ability to share data, analysis, and features between systems has become a large problem in the Industrial Internet age. All systems and machines need to communicate in a common set of services based semantic containers to facilitate data exchange and system and business level analytics.

GE's concept of a Brilliant Factory affects three major areas of the value chain: (1) design engineering, (2) manufacturing systems operations and (3) service operations. The basic concept is to create a digital thread from design to product engineering to manufacturing engineering to manufacturing and supply chain operation all the way to services closing the loop back to engineering. Such a digital thread will enable shorter learning cycles between manufacturing and design, leading to faster product development and flawless factory launches. The process starts with design, which results in a digital blueprint. And once a design has been determined, it is transmitted digitally to manufacturing engineering. In addition, factory flow and layout, robots and manufacturing controls will be simulated and optimized prior to actual production.

A software architecture and underlying enterprise system needs to be created to handle the data exchange at a very low level, internal to design, manufacturing, and services domains and facilitate movement of data between these domains. This can be thought of as an internal data thread and an external services infrastructure.

Challenge:

GE is driving towards the goal of a connected Industrial Internet, this requires a highly scalable and structured network with expandable data processing and storage infrastructure. The complexity of the data movement and filtering and data network requirements dictates the need for a system that can distribute the complex analytics and data to the appropriate subsystem.

An example of a complete system of a large business problem would be an energy company trying to operate and optimize a fleet of wind farms. The pattern is that to optimize a farm of wind turbines, there is a need for machine level high frequency data at the turbine to understand and model wind directions and effects on individual turbines. The system then needs the answers to these analytics at a lower frequency to predict at a farm level the wind effects on upstream and downstream turbines. To try and gather all the high frequency data into a single system would quickly overwhelm the network and storage mechanisms.

The fleet level analytics needs visibility into individual farms to understand the optimal operating parameters based on annual electricity production (AEP). The data frequency is even less in that the prediction of these systems is a longer-term problem and less frequent data is needed to see the macro trends of all system.

The combination of the need to have very accurate answers at the edge, combined with analytics at a fleet level to control specific farm variable, and combined with a long term strategy of business or fleet level predictors requires a system that can distribute complex analytics to any component in the system. In short the platform needs to be able to encapsulate small parts of analytics and have the flexibility to move the compute to the data.

The Architecture:

The current industry platform architecture for current design of this system is a distributed enterprise architecture that pulls all the data and analytics to large data warehouse. This solution ignores the power of distributed and compartmentalized computing and the scale of the edge in favor of a more command and control architecture.

The concept of small agile blocks of analytic code are not new, they have been part of object oriented programing for some time. The increase in the size of data collected and the focus on an Industrial Internet and Internet of Things requires a new paradigm in how we collect, pre and post process data for analytic results.

The current services oriented architectures rely on the systems ability to quickly access large pre-processed data stores, for instance Hadoop clusters require a significant effort to pre-process data to get it into the proper form for analysis. This central data store pattern requires in most cases duplication of data and storage. What would a system architecture look like that could easily transfer parts of end analytics to the edge where data is being recorded to pre-process incoming data to answers that would be required by the next level of processing.

The concept of analytics processing raw data and returning an answer becomes key in a connected system because an answer is almost always smaller that the raw data. This pattern of processing data close to its source becomes very powerful in the next wave of enterprise systems needed to fuel the Industrial Internet. The need for an architecture that can distribute compute and orchestrate small parts of analytics becomes paramount in guaranteeing scale and the ability to distribute the compute load.

Microservices:

The idea of Microservice Architecture as a way to design software as a set of suites of independently deployable software units lends itself well to the needs of GE's Brilliant Factory initiative.

Microservices is an approach to develop a set of small services that can be orchestrated into a larger application solution. The ability of this architecture to distribute, scale, and secure various components of the system play a key role in being able to build large cross business solutions. To explain the Microservice style it's useful to compare it to the monolithic style: a monolithic application built as a single unit. Enterprise Applications are often built in three main parts: a client-side user interface (consisting of HTML pages and JavaScript running in a browser on the user's machine) a database (consisting of many tables inserted into a common, and usually relational, database management system), and a server-side application.[i] The server-side application will handle HTTP requests, execute domain logic, retrieve and update data from the database, and select and populate HTML views to be sent to the browser. This server-side application is a *monolith* - a single logical executable.[ii] Any changes to the system involve building and deploying a new version of the server-side application. [iii]

GE is actually a business of businesses, each having its own requirements, systems, and goals. Creating a single platform to serve all systems and businesses is not attainable in our lifetime. A system that will enable short term needs and plan for long term goals is needed to ensure future success. Microservices allow enterprise architects to have a base to build out their low level building blocks that overtime will be orchestrated and collected into large business solutions.

Questions:

- What are the definitions of business capabilities of Microservices approaches?
- Can Microservices address multiple levels of business abstractions?
- How well do Microservices scale and distribute compute, does the design concepts included answer these concerns?
- How do we ensure standards across multiple businesses and internal domains?
- Is there an aspect of Microservices that addresses semantics of the service end points?
- Does Microservices address legacy software (i.e. 40 year old Fortran codes) through encapsulation?
- How does Microservices handle large 3rd party engineering solutions?
-

---

[i] Martin Fowler: Microservices "http://martinfowler.com/articles/microservices.html"
[ii] The term monolith has been in use by the Unix community for some time. It appears in The Art of Unix Programming to describe systems that get too big.
[iii] Martin Fowler: Microservices "http://martinfowler.com/articles/microservices.html"