The Power and Monetization of Microservices

Posted by Nicholas Milani (https://blogs.saphana.com/author/nicholas-milani/) on May 31, 2017

More by this author > (https://blogs.saphana.com/author/nicholas-milani/)

Tags: abacus (https://blogs.saphana.com/tag/abacus/) cloud foundry (https://blogs.saphana.com/tag/cloud-foundry/) Cloud Platform (https://blogs.saphana.com/tag/cloud-platform/) Hybris (https://blogs.saphana.com/tag/hybris/) microservices (https://blogs.saphana.com/tag/microservices/) organizing for innovation (https://blogs.saphana.com/tag/organizing-for-innovation/) revenue cloud (https://blogs.saphana.com/tag/revenue-cloud/) sap cloud (https://blogs.saphana.com/tag/sap-cloud/) scp (https://blogs.saphana.com/tag/scp/) yaas (https://blogs.saphana.com/tag/yaas/)

(https://www.facebook.com/sharer.php?u=https%3A%2F%2Fblogs.saphana.com%2F2017%2F05%2F31%2Fpower-monetization-microservices%2F&t=The%20Power%20and%20Monetization%20of%20Microservices)

(https://twitter.com/share?text=The%20Power%20and%20Monetization%20of%20Microservices&url=https%3A%2F%2Fblogs.saphana.com%2F2017%2F05%2F31%2Fpower-monetization-microservices%2F)

(https://plus.google.com/share?url=https%3A%2F%2Fblogs.saphana.com%2F2017%2F05%2F31%2Fpower-monetization-microservices%2F)

(https://www.linkedin.com/shareArticle?mini=true&url=https%3A%2F%2Fblogs.saphana.com%2F2017%2F05%2F31%2Fpower-monetization-microservices%2F&title=The%20Power%20and%20Monetization%20of%20Microservices)

Large enterprises are often terribly inefficient. Software companies are no exception. As a company's product portfolio expands, so does the potential for redundancies and inefficiencies in what is developed. Acquisitions bring forth new technology stacks, and 're-platforming' typically delays an innovation roadmap for many quarters. Concepts around master data, integration, and other key processes can be inconsistent and may overlap, or can even be re-produced and re-imagined repeatedly.

What if internal development teams could be corralled to eliminate duplicate effort, realize massive economies of scale, and increase efficiency? What if this efficiency could result in lower development costs, quicker times to market, and a burgeoning ecosystem of work that operationalizes and monetizes much of your core intellectual property?

Microservices —a modern architectural style in use by a growing number of large software companies— is the proven way forward to meet these needs. Microservices are small, self-contained services that can be brought together to create a business or consumer application. Self-organizing and autonomous teams can easily build out needed application features by leveraging services produced by other teams. Updates can be rapidly pushed out at a granular level with little disruption to customers. Spotting the services and features that are being used across the enterprise versus ones that should be retired also becomes a simpler task.

By building robust sets of microservices a wealth of capabilities can be easily consumed across the entire enterprise. Core concepts for key areas such as customer, order, and product can be managed in a consistent way. The value add and rate of innovation brought on by a microservices ecosystem is exponential. And combining these services on a platform where developers, customers and partners can create, extend and monetize their work takes application development and innovation to a whole new level.

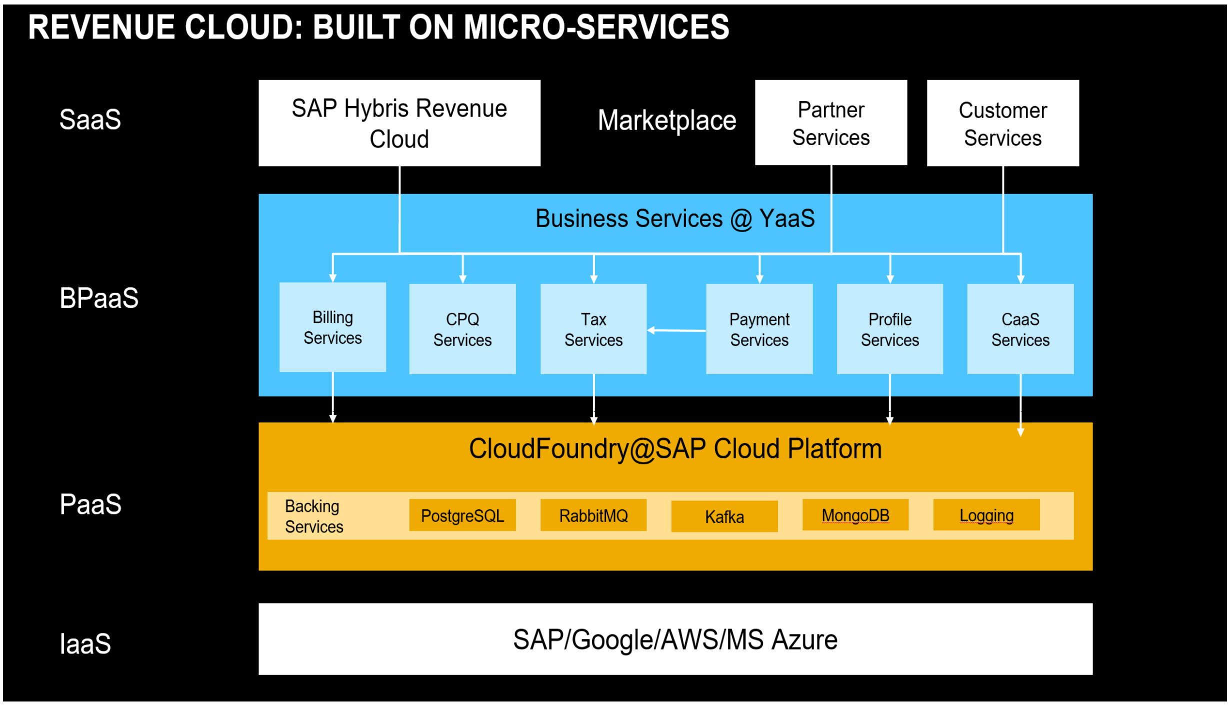
The key to a successful microservices economy is of course stellar content, but also adequate compensation for contributors – the developer teams producing services for others to consume. An application that is purchased by an end customer may leverage services built by dozens of different teams, but the complexity of cost and pricing should be handled automatically behind the scenes. The way this is typically handled is to combine service subscriptions with usage-based pricing, revenue share, and settlement models.

With potentially multiple teams attributed to every transaction, the rules can be complex requiring multiple levels of calculation before compensation can be totaled. Payments and settlements can be made in a variety of ways, including measuring and tracking against a key set of KPIs instead of just monetary compensation, depending on what the service developers are being measured on.

I have seen first-hand how this type of ecosystem works with the SAP Hybris Revenue Cloud (http://hybris.com/en/products/billing/revenue-cloud) application development team. Based on a micro-services architecture, large parts of the app's capabilities are being built by simply tapping into customer management services, payment services, and tax services created by other internal SAP teams. Internal pricing for services are published in a way that each team can model their service's usage into operational cost calculations. The services are then subscribed to and metered by the underlying Abacus and CloudFoundry platform.

Leveraging other teams' services shaved months off the app's development efforts, and the teams that built the services the app consumes receive compensation for their work. By appropriately rewarding these teams, they are incented to continually enhance and build new services.

The Revenue Cloud app also provides dozens of microservices which can be consumed to complement other applications. Easy integration of partner or other custom services is on the horizon, which will provide customers with new and exciting capabilities faster than ever before. Revenue Cloud is also being designed to handle the monetization of this internal Microservice ecosystem.



REVENUE CLOUD: BUILT ON MICRO-SERVICES

SaaS

SAP Hybris Revenue Cloud

Marketplace

Partner Services

Customer Services

BPaaS

Business Services @ YaaS

Billing Services

CPQ Services

Tax Services

Payment Services

Profile Services

CaaS Services

PaaS

CloudFoundry@SAP Cloud Platform

Backing Services

PostgreSQL

RabbitMQ

Kafka

MongoDB

Logging

IaaS

SAP/Google/AWS/MS Azure

SAP Hybris Revenue Cloud architecture, featuring microservices

By harnessing the power of previously isolated teams and siloed capabilities, while operationalizing ways to properly compensate all the teams in the ecosystem, exponential benefits can be realized through the use of microservices architecture.

To learn more about “Monetizing your applications using Cloud Foundry Abacus and Billing Engine please join Nicholas Milani and Pankaj Kumar at Cloud Foundry Summit, Santa Clara on June 14th at 11:30AM. (https://www.cloudfoundry.org/event/summit-silicon-valley-2017/)

Average User Rating

Rating: 0.0/**5** (0 votes cast)

1397 Views

❮ Previous (https://blogs.saphana.com/2017/05/24/better-place-better-time-now-thoughts-orlando/)

Next ❯ (https://blogs.saphana.com/2017/06/01/digital-transformation-and-the-data-warehouse/)

Leave a Reply

You must be logged in (https://blogs.saphana.com/wp-login.php?redirect_to=https%3A%2F%2Fblogs.saphana.com%2F2017%2F05%2F31%2Fpower-monetization-microservices%2F) to post a comment.