WHITE PAPER

Embracing Microservices Architecture in Telecommunications

Adoption and measuring success of microservices

Infosys
be more

The telecommunications industry is in a transition phase. Physical networks are being replaced by digital ones. Moving beyond communications services, the sector is now providing industry-specific technology solutions. Sales, support, and delivery operations are transforming from a simple solo mode to a complex collaborative one.

The industry is exploring business, IT, and network strategies to manage the changing business environment. Focusing on improving the customer experience, telcos are revising their IT and network architectural capabilities to support extensibility and elasticity. A key mechanism for enabling these strategies is adoption of microservices architecture (MSA).

Adopting MSA architecture in telecommunications will not only be a technology transformation but also a major shift in people's mindsets, the organizational culture, and team structures. While microservices architecture will ensure flexibility and scalability, its implementation needs to be meticulously planned and executed. Microservices development goes hand in hand with DevOps-based delivery model to deploy business capabilities which need agility and on-demand scalability.

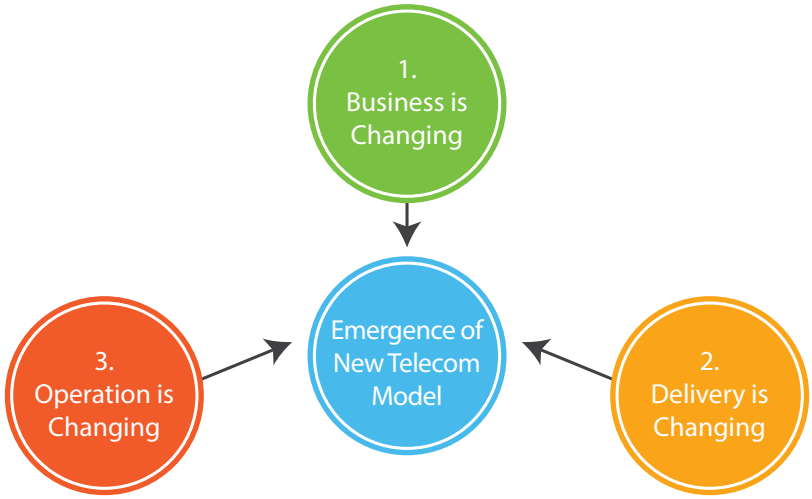
Since implementation of the architecture will need expensive foundational blocks such as establishment of DevOps practices, automated delivery platform, and complex infrastructure operation management, it should be continuously monitored to ensure efficient and effective implementation of the MSA strategy. A well-designed MSA monitoring strategy will be required to keep its governing body informed about any deviation from the planned path.

This point of view outlines the rapidly changing telecommunications industry and how the microservices architecture strategy can help the industry respond to new challenges. It focuses on:

1. How communications service providers (CSPs) can adopt microservices architecture?
2. How CSPs can monitor the effectiveness of the MSA strategy?

The telecommunications industry is changing

The telecommunications industry will be transitioning from its business path to enter uncharted territory. It will explore collaborative business models by venturing into industry-specific offerings. It could be 'uberising' its network by moving it from the ground to the cloud. It should prepare to move into a service guarantee-based multi-partner delivery and support ecosystem. The customer will be the center of innovation and browse-first will become the norm.



Figure#1 Telecommunications Industry is changing

The business change

The collision of telecom products and services with other industry domains is providing CSPs with the opportunity to explore new technology service offerings. The evolution of a new telecom service ecosystem is restructuring the organization e.g. new service lines; redefining the business process e.g. sales, support and revenue models; and more automation of business operations e.g. service fulfilment orchestration with multiple partners.

The delivery change

The adoption of software-as-a-service (SaaS), platform-as-a-service (PaaS) and cloud technologies is transforming the telecoms industry's core assets - the physical networks - into digital e.g. software-defined networking (SDN) and virtual network function (VNF). It is supporting an emerging demand for every operation to be on-demand, intuitive and intelligent e.g. self-service, zero touch provisioning, self-healing and real-time analytics.

The operations change

The telecommunications industry, as a digital enabler of other industry offerings, needs a platform to support a service level agreement (SLA) based operational model. Signing SLA-based operational models with industry partners will compel CSPs to reorganize themselves with service-level objective (SLO) based internal operational functions e.g. service desk management shifting from reactive incident resolver to proactive problem finder and solver.

The imperative

The business, delivery, and operational changes need enabling applications to be responsive to these changes. Telcos will need IT and network systems that can be continuously augmented with new business and operational capabilities. They will need scalable applications for processing millions of events, requests, and transactions generated from automated provisioning and operational activities. They will require more business and operational process automation to ensure reliable time-to-deliver, time-to-support and time-to-invoice the services.

The challenge

In today's 'experiment-and-explore' business model, the rate at which system requirements change is outpacing commercial off-the-shelf (COTS) product release cycles. Most COTS product vendors and bespoke application developers are experiencing technical and delivery constraints to deal with continuous change, processing spikes and assuring business process reliability. Other industries have successfully responded to these challenges by moving key business capabilities from COTS to SaaS packages, adopting continuous integration (CI) and continuous delivery (CD) release cycles and decomposing monolith applications into discrete microservices.

Microservices in telecommunications

The technology sector has successfully applied MSA to overcome challenges of elongated release cycles and inefficient horizontal hardware cloning-based system scaling. Now, this proven approach is gradually expanding to the telecoms industry.

Microservices are redesigning BSS

The critical components in the business support systems (BSS) domain such as sales order capture, commercial order management, product catalogue and offer management are good candidates to adopt microservices. Adoption of microservices architecture in BSS can protect expensive customization and major upgrades of COTS packages. The MSA allows you to decouple the 'system of engagements' from the 'system of records', and gives agility for differentiation in a hypercompetitive market. The missing capabilities in BSS can be implemented as microservices on a platform-as-a-service (PaaS). It will reduce time-to-market for new service launches, and will allow management of seasonal scalability requirements. There are different patterns of microservices that can be applied:

- **Discrete microservices** – They are developed to support unique business capabilities, and good candidates to implement discrete microservices are product / catalogue , offer / promotions, payment.

- **Pass-through microservices** – They are developed to support legacy system integrations, e.g., Tuxedo integration with mainframe systems.

- **Orchestration microservices** – They are developed to support workflows and calls of multiple discrete microservices. Good candidates for implementing orchestration microservices are cart and order orchestration.

Microservices are redesigning OSS

MSA provides capabilities for operations support system (OSS) components to manage frequent changes to individual network interfacing components. Discrete microservices are good candidates to decouple the OSS from constantly changing environments. Some of the processes where microservices pattern can be applied are network event monitoring, network fault alarms, technical order decomposition, inventory provisioning and resource activation. It can provide agility for OSS software upgrades and stability in dynamic interfaces with heterogeneous network element managers.

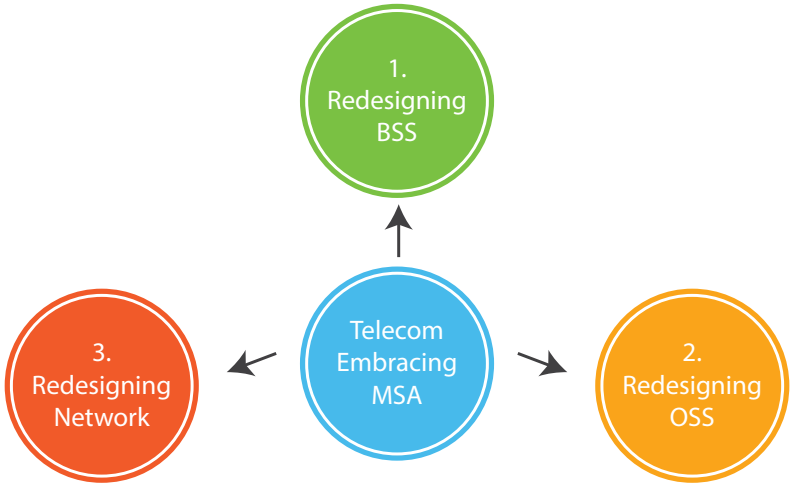


Figure #2: Microservices in telecommunications

Microservices are redesigning the network

The relevance of MSA for telecommunications is not limited to BSS/OSS IT, instead it will be gradually moving deep down into network management, network elements, and functions.

Telecom network technology vendors have already started seeing the relevance and benefits of MSA alignment, and many have announced strategies to redesign their network orchestration and management platforms.

The virtual network function will eventually move from virtualized in-premise commodity hardware to the cloud, and the cloud-native VNFs can be deployed as microservices. The MSA-aligned VNFs will simplify feature upgrades and have virtually unlimited scalability. They will support dynamic provisioning.

Microservices-based building blocks are used in software-defined network platforms to create dynamic and on-demand 'bandwidth' services. The SDN control functions - designed and deployed as microservices - can continuously be augmented with new capabilities. It also provides scalability required by OSS systems and other third-party applications communicating with the SDN platform.

Microservices are moving (not removing) the complexities

MSA will move complexity from software 'design-build-test' implementation process to automated 'service routing-messaging-monitoring' operation management process.

Microservices will increase developer productivity by limiting their scope to a smaller set of cohesive features without being concerned about impact on other services. But CSPs must ensure meticulous design and also be ready to make significant investments in automation tools for operational support and maintenance.

CSPs should be aware that decomposing BSS/OSS and network services into microservices will offer greater flexibility, but communication complexities among them could bring performance challenges too.

As microservices support quicker delivery and runtime flexibility, CSPs need to change their funding from fixed cost to continuous business benefit-based investments.

The biggest impact CSPs would need to manage is organizational change. They will have to reorganize their teams from technology to functional/feature domain. The feature teams should have skills to design, develop, test, deploy, operate and maintain.

Strategy to adopt microservices architecture

Introducing microservices architecture for telcos is a gradual and iterative process for which a long-term strategy has to be formulated. To make the most of the MSA pattern, CSPs must first build the foundation, then discover, design and develop relevant microservices use cases.

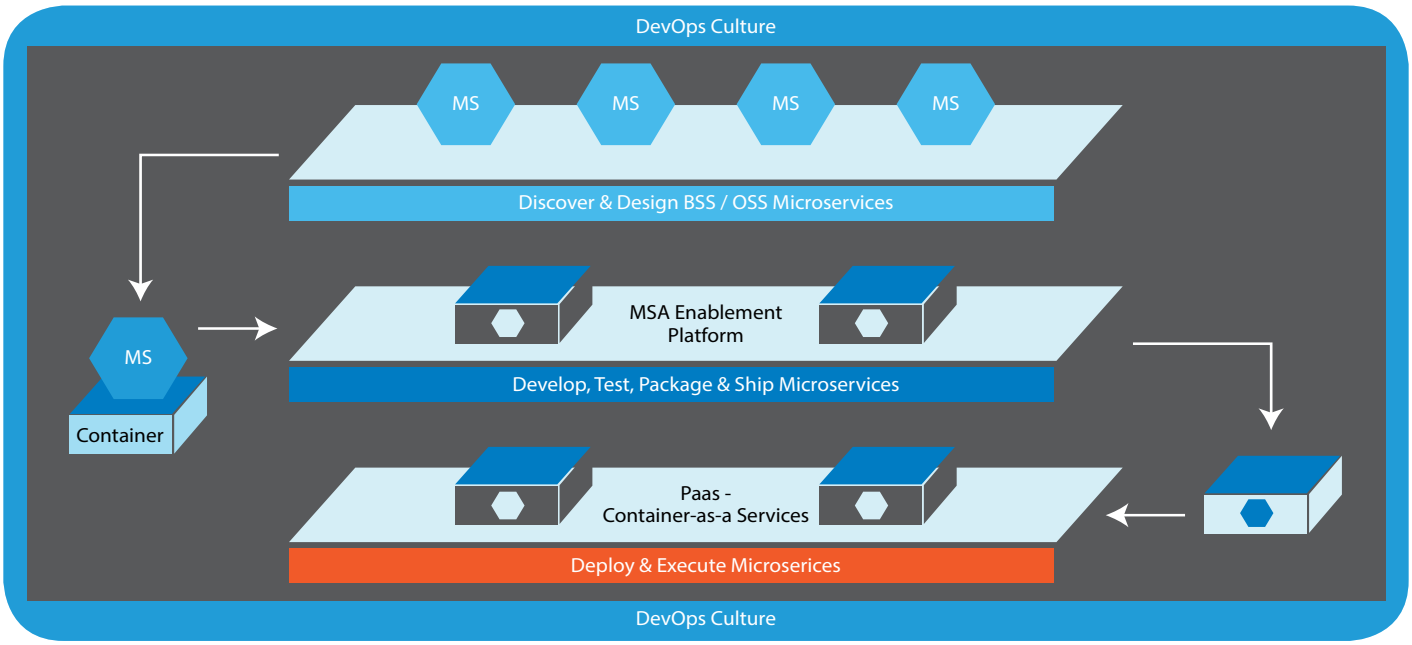


Figure #3: Microservices adoption strategy

Change the culture

The main problem that the DevOps, automated delivery and MSA together are expected to solve is welcome change requests. It is a cultural shift from opposing change to embracing change. The DevOps culture supports constant change, i.e. continuous development, integration, and delivery. DevOps brings a sense of ownership and collaboration across change enabler entities – operator, designers, developers, and testers.

The DevOps practice can help persuade the CIO to accept fluid change management, but may not be sustainable in a complex collaboration environment. An automated delivery platform for build, test, and deploy will be needed to move user stories from developer to consumer desks in frequent release cycles. It will give flexibility to developer communities to execute their tasks frequently and repeatedly without manual intervention.

The microservices architecture pattern exhibits many characteristics which make it perfect for agility and automated delivery. A microservice represents one self-governing business operation. It makes development, testing, and deployment simple. It has hard boundaries with minimal or no dependencies. This encourages frequent changes with limited or no ripple effects. It exposes its service through a well-defined service contract – the open interface. It simplifies the integration and composition of services.

Prefer the PaaS

The second important problem that a microservices architecture pattern aims to solve is to face volatile volume. It will need an elastic hosting infrastructure and platform to address dynamic scaling demand. In contrast with hardware virtualized infrastructure, the OS virtualized container technologies offer more elasticity with much lower virtualization overheads.

Microservices architecture's semi-autonomous characteristic makes it a natural fit for containerization. A container encapsulates a microservice with its complete runtime dependencies such as libraries, binaries and config files. Containers can be instantiated and ceased in within seconds because there is no guest OS to be started or shut down. The containers are resource consumption-aware. So microservices deployed in containers are best suited for elastic scalability.

When the container manager sees the surge in demand for a microservice, it can replicate the microservices by instantiating more containers. It makes applications vertically (Y-axis) scalable. Furthermore, it also tracks resource utilization and distributes the load among the servers. This gives horizontal (X-axis) scalability.

Microservices architecture simplifies development because its scope includes bounded context (the features with minimum or no dependencies) within a given context in the problem domain (the function). Applications comprising different microservices need to be

eventually managed as one application. A multi-container-based application requires a container management layer over the infrastructure.

Building a custom environment for containers' auto scaling and load balancing will be a mammoth task. It is more feasible to rent a PaaS emulating container-as-a-service that can manage orchestration, scheduling and clustering of containers distributed across nodes.

Discover the microservices

Once the CSPs successfully institutionalize the DevOps culture and arrive at a consensus to move capabilities from static infrastructure to dynamic PaaS, they should be ready to scan their application capabilities landscape and discover opportunities for microservices application.

There are many methods to discover and design microservices, including the most practiced domain-driven design using noun (entity) based and verb (activity) based decomposition. Whatever method is used, a microservice should represent a discrete business operational activity. A well-choreographed microservice will form a business operational function. When these functions are orchestrated through a workflow, it will eventually reflect the end-to-end business process.

The microservice could be the base unit of the CSP's business process. The following diagram illustrates an approach to identify the candidates for microservices in the order-to-cash process.

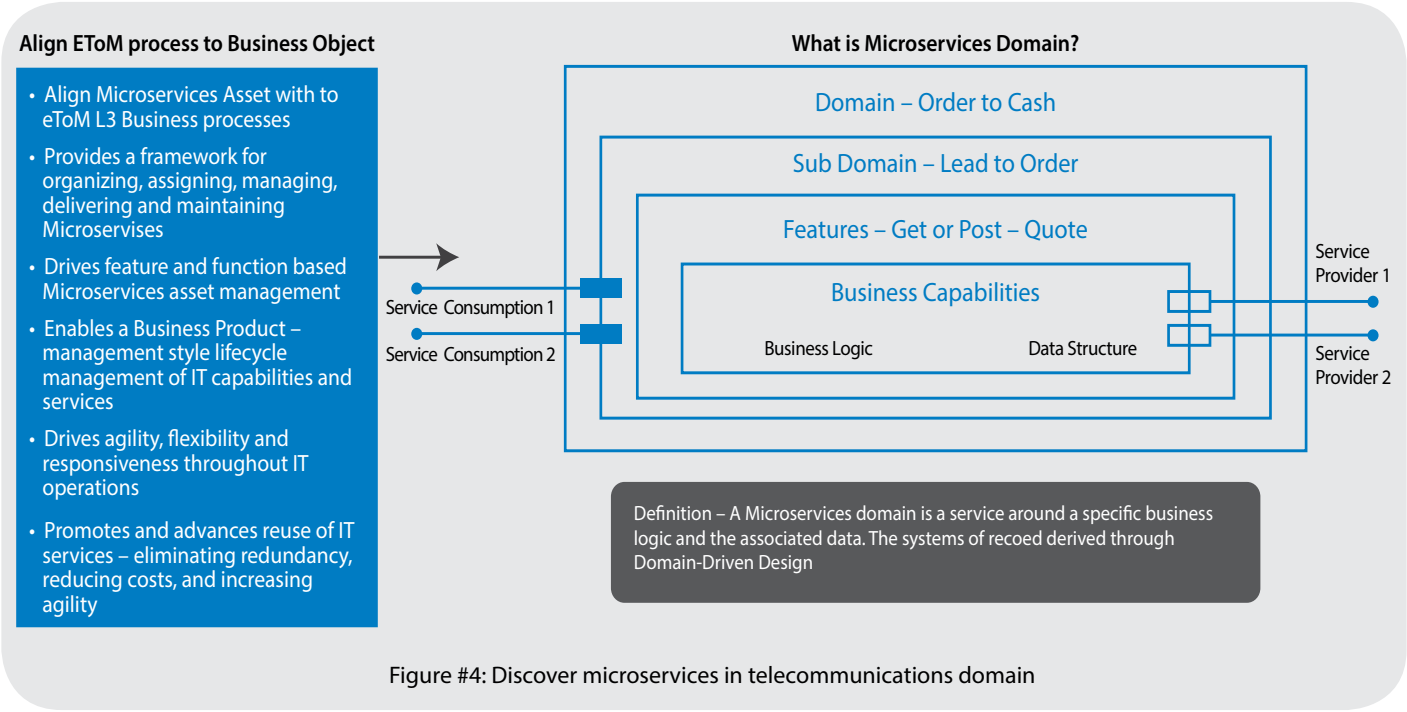
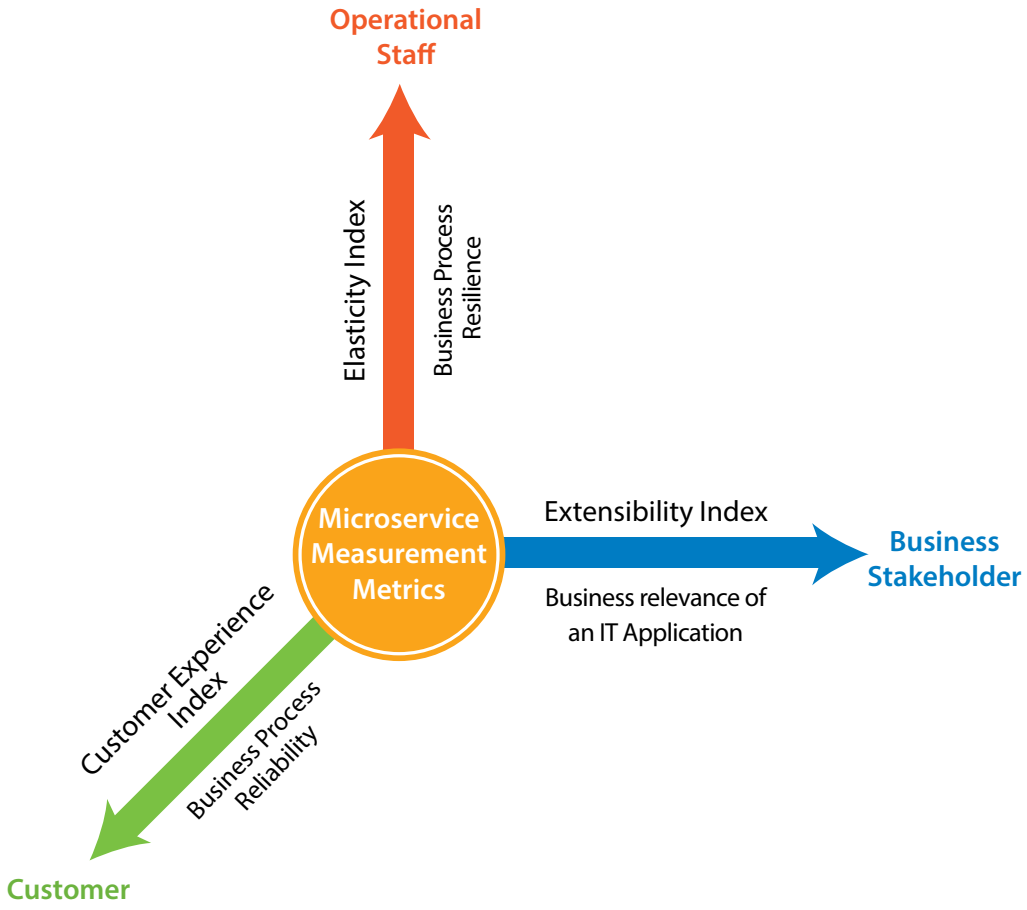


Figure #4: Discover microservices in telecommunications domain

To summarize, microservice adoption is an iterative (not big bang) journey. The CSPs need to define and follow their multi-year iterative microservice transformation road map. The journey begins with establishing the foundation with DevOps and PaaS preference, and then continues with a dynamic iterative approach for adding new and shifting existing business capabilities. The TM Forum's eTOM Level 3 business processes could be a good reference point to discover microservices based business capabilities across sub-domains of the telecom service value chain i.e. concept to market, prospect to order, order to activate, trouble to resolve and usage to cash.

Strategy to monitor microservices effectiveness

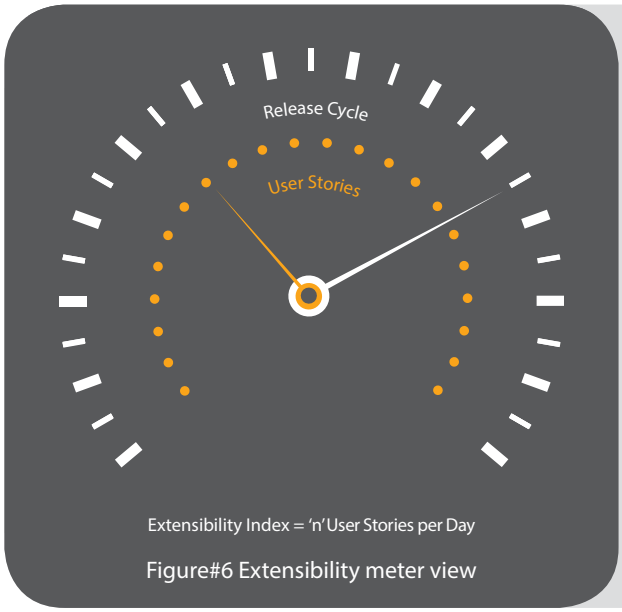
Once the microservices business and operational capabilities deployments begin in production, a governance body driving MSA programs will need a microservice navigation compass. It is critical to ensure alignment of IT programs with expected business outcomes. CSPs should design a navigation compass with three-dimensional measurement metrics that can continuously monitor their microservices strategy effectiveness across business stakeholders, operational staff and most important - the customer.



Figure#5 – Microservices navigation compass – Business measurement metrics

Extensibility meter

The extensibility meter measures 'rate of change'. It offers information about the number of capabilities (user stories) changed or augmented in an application during a release cycle.



Figure#6 Extensibility meter view

- Release cycle (C_t) represents time required to deliver a user story into system capability
- User story (U_s) is a business requirement statement with acceptance criteria

$C_t = f(\Sigma U_s)$
Release cycle is direct function of number of user stories.

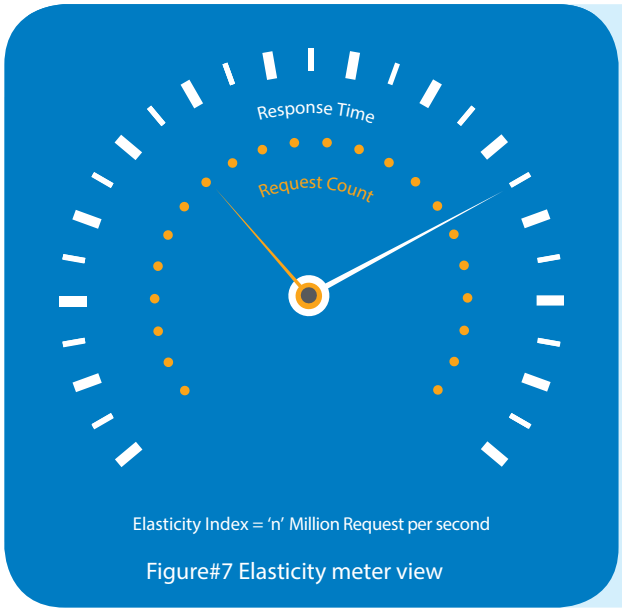
Extensibility Index = $f(1/C_t)$
Extensibility is inverse function to release cycle time.

How to interpret extensibility reading

Higher extensibility means a shorter time for a business user story to be available in a production application, keeping the application relevant for business stakeholders at all times. It indicates the degree of application relevance with reference to business stakeholders' demand.

Elasticity meter

The elasticity meter measures response time. It offers information about the number of requests that a system can process in a given time period i.e. acceptable response time.



Figure#7 Elasticity meter view

- Response time (R_t) represents time taken by a service to accept a request, process a request and provide a response
- Request (R_q) is an operation that a consumer service wants the provider service to perform

$R_t = F(\Sigma R_q)$
Response time is a direct function of number of requests.

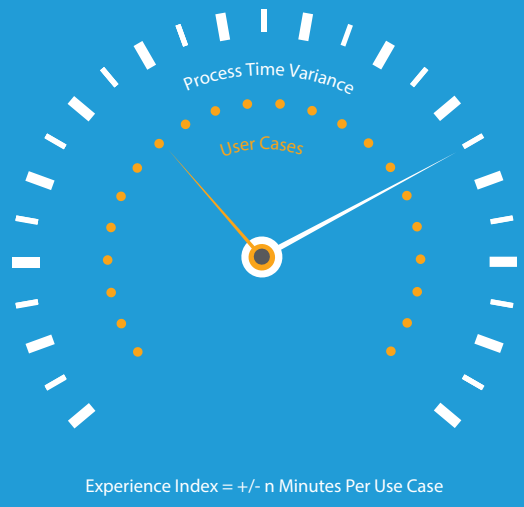
Elasticity index = $F(1/R_t)$
Elasticity is inverse function of response time.

How to interpret elasticity reading

More elasticity means that a system design is highly capable to scale in a very short time to meet any surge in requests without impacting the response time. It involves maintaining a resilient business operation in an ultra-high data volume band with fluctuating demand scenarios. It predicts the probability of meeting or missing SLA/SLO by business operations.

Experience meter

The experience meter measures business process 'cycle-time variance.' It indicates business process reliability to complete a customer transaction within a given time period i.e. committed customer time.



Figure#8 Experience Meter View

- Process variance time (P_t) indicates time variance of a process to execute the same process with different variables
- Use cases (U_c) represent a scenarios with different input variables

$P_t = f\ (\Sigma U_c)$
Process variance time is direct function of number of use cases.

Experience Index = $f\ (1/P_t)$
Experience reliability is the inverse function of process variance time.

How to interpret experience reading

A high experience reliability reading shows business processes are automated to an extent where the cycle time is consistently repeatable and reproducible in different scenarios (various combinations of customer segment, products, transaction volume). It is the degree of consistency in offering a standardized customer experience.

Role of CSP's IT Partner

A microservices architecture strategy can be a catalyst for a business transformation.

CSPs must partner with an IT services company for their microservices adoption road map. It can help discover relevant business capabilities (benefits) and manage increased operational complexity.

DevOps, as an enabler of microservices adoption, is more about bringing cultural change than technology change. The CSP's IT service partner can support in establishing process, people and platform change management required for institutionalizing DevOps culture.

Microservices are relevant for business alignment and business ownership of IT. The IT service provider can help bring IT and business alignment by monitoring, measuring and guiding the CSP's microservices implementation success.

A CSP should partners with an IT services company for return on investment and risk mitigation in MSA.

About the author

Sanjay Kumar Verma

Principal Consultant (Telecommunications), Infosys

Sanjay is a Business Solution Consultant, with over 18 years of experience in Telecommunication Industry across BSS and OSS applications. In his recent engagements, he has played advisory roles at CxO level for drafting IT Architecture roadmap, technology evaluations and solution definitions. He is passionate about analysing relevance of trending technologies for Telecom industry and how it can be leveraged to realize his client's business strategies. He has published thought papers on – SDN/ NFV, APIs and Microservices.

Sanjay can be reached at SanjayKumar_Verma@infosys.com

For more information, contact askus@infosys.com

Infosys be more