FACT SHEET

SERVERLESS I.T. CONSULTING

# OUTMANEUVER THE COMPETITION WITH GAME CHANGING I.T.

CLOUD-BASED, STACK-FREE AND SERVER-LESS I.T. FUNCTIONS AND APPLICATIONS

## SERVERLESS I.T. CONSULTING

- ❑ MICROSERVICE-BASED ARCHITECTURES
- ❑ DOCKER/CONTAINERIZED BASED ARCHITECTURES
- ❑ OFF-PREMISE AUGMENTATION ARCHITECTURES
- ❑ STACK-FREE & SERVER-LESS ARCHITECTURES
- ❑ WEBSERVICE / API-BASED APPLICATIONS

nmv 1.5

## WHAT CAN YOU ACHIEVE WITH SERVERLESS I.T.?

FACT SHEET

To remain competitive in business climates that change at lightning speed, **nimble technology** adoption must be a strategic priority. Businesses can no longer afford to make physical technology investments that limit their ability to adapt. Fortunately there's a better way, and our **SERVERLESS PRACTICE** can take you there.

The legacy of purpose-driven infrastructure purchases destined for corporate datacenters now has a superior alternative: Stack and server-free cloud **microservices** and **programmable functions** that can be snapped together using **Event** and **Service Oriented Architectures** to achieve the same results; but with better technical and business characteristics. Combining these ideas with **containerized** approaches, seasoned professionals with decades of datacenter experience deconstruct your operation to determine where refactoring I.T. functions or applications onto these modern alternatives, can achieve the advantages that you seek. Starting with an *enterprise* or *line-of-business* scoped assessment, we catalog opportunities for I.T. agility improvements. Each opportunity includes a recommendation on how it's function or application can be re-platformed, for optimal results. We then provide developer consulting to help re-platform services onto these cloud-based, agile alternatives.

# SERVERLESS I.T. CONSULTING

Continuous B2B and B2C innovation pressures means I.T. functions and applications must be change-ready! Whether for new or legacy systems, we help I.T. become more agile with **lift-and-shift**, **containerization** and **refactoring** approaches to cloud supplementation. Building blocks include:

### ❑ MICROSERVICES BASED ARCHITECTURES

Micro-service architecture is a SOA approach to *developing an application as a suite of small, independent*, *functional services*; each running in its own process and communicating via lightweight mechanisms (often a HTTP resource API). Each of these services is built around a well-defined/well-encapsulated business capability that that service is uniquely responsible for; and all such services are deployable independent of one another. Using a catalog of such black-box "microservices" means that developer and DevOps resources can focus strictly on features of the *composed* application.

### ❑ STACK-FREE & SERVER-LESS ARCHITECTURES

Stack-free and Server-less architectures refer to applications that are built on services known as Backend-as-a-Service; or on custom code that's run in ephemeral Function-as-a-Service containers. The best-known vendor for this implementation paradigm is AWS' Lambda service. Using these ideas, application behavior is shifted to the front-end, resulting in architectures that remove the need for traditional *always-on* server systems. Depending on the use-case, such deployments can significantly improve I.T. agility, reduce operational cost and complexity; but at a tradeoff of vendor dependency adoption.

### ❑ DOCKER / CONTAINERIZED BASED ARCHITECTURES

Containerized server architectures use traditional *always-on* servers, however the guest operating system and/or application is wholly contained in it's own O/S level, lightweight virtual environment; complete with required dependencies. Containers, whether Docker or Linux/LXC based, permit black-box development of reusable features, permit higher server utilization, and improve service mobility.

### ❑ WEBSERVICE / API BASED APPLICATIONS

WebService applications are applications written to programmatically expose their functionality to other applications via a API; typically HTTP method based. They allow applications developed in different technologies to communicate with one another via common exchange formats, such as XML and JSON. They are not tied to any one operating system or programming language and may, on the back-end, employ any of the aforementioned building blocks.

### ❑ OFF-PREMISE AUGMENTATION ARCHITECTURES

Increased I.T. agility can be achieved even without refactoring functions and applications to event/S.O.A.-based services. BC/DR solutions can, for example, temporarily rent compute, storage and network infrastructure resources necessary for the continuance of mission-critical business during outages; thereby avoiding capital outlay.

PRISMALYTICS io count·on·it

SERVERLESS I.T. CONSULTING for better business outcomes!
**web**: www.prismalytics.io **email**: connect@prismalytics.io

nmv 1.5