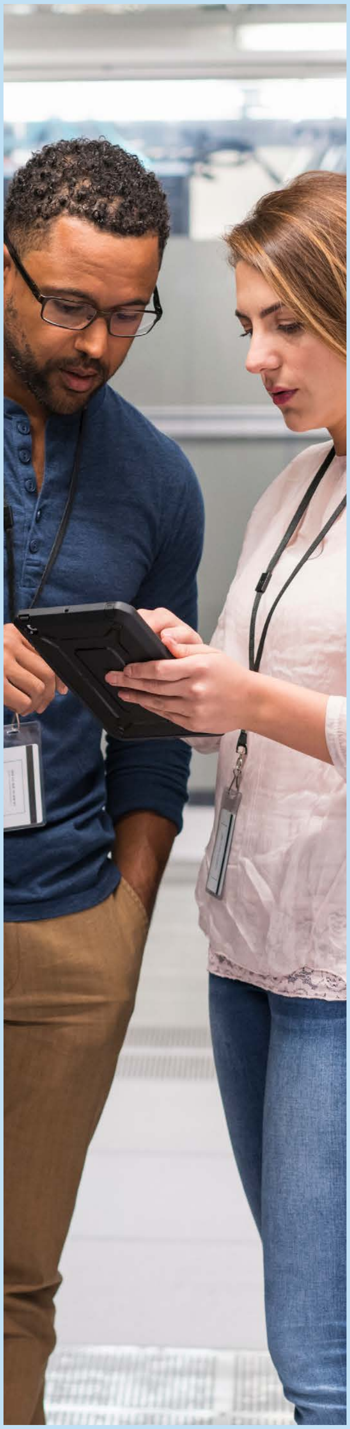
VIEW POINT



SERVERLESS IN DIGITAL LANDSCAPE AN INFOSYS POINT OF VIEW

By Archa Shukla

Infosys®
Navigate your next

Executive Summary

Digital landscape is ever growing and evolving in order to meet the changing market demands. Internet of Things, Blockchain, Augmented Reality, Virtual Reality, Robotic Process Automation and other advanced technologies are shaping the future of enterprises and driving the need for a Digital transformation work. The real time information provided by IoT is being utilized in many Mobile, Web applications and is deployed in almost all the verticals namely Energy, Telecom, Agriculture, Manufacturing, Health and Retail. More and More customers want to be digitally connected using the latest technologies and this is creating a need for dynamically scaling the Enterprises Digital Landscape. Enterprises today with digital offerings see an ever-increasing demand for agile development environment, high scalability and availability to meet the ever-increasing Customer demands for a digitally connected world.

In this context, we present our point of view that addresses this current need and explains how Serverless architecture can help in modernizing the current digital landscape of an Enterprise and play a vital role in its Digitalization journey. Serverless should be integrated as a core part in any Enterprise Digital Strategy . We suggest the use of Serverless in building Customer Experience Platform and extend the discussion further to embed IoT data in the landscape for a complete connected user experience. We present the reference architecture and some of the complex use cases where Serverless architecture can benefit the Enterprise and help it address the need to quickly scale the systems for rapidly increasing load by ever increasing Consumer base of digital services.

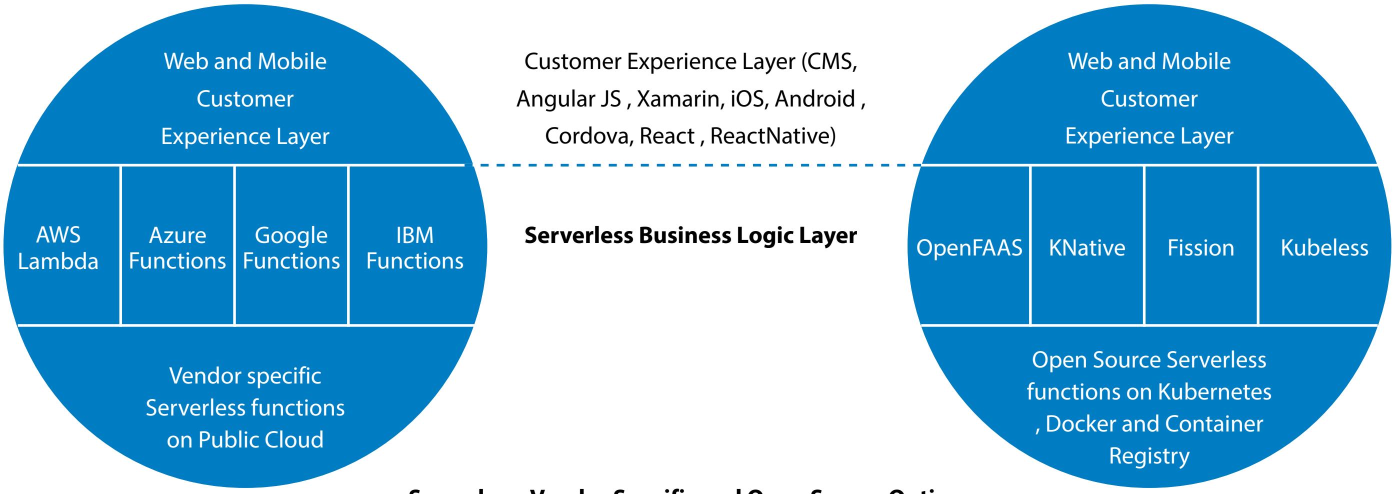
Introduction

Architecture of present day leading edge applications need different ways of data interaction and requires more adaptive and dynamic capabilities. Hence, the need of the hour is to move these implementations from a monolithic architecture to a Micro service based architecture that is event driven and more data centric. Functions as a Service approach take micro services approach to next level by providing smaller lightweight, independently scalable and fully managed functions as services **(Functions as a Service)** that are **Event Driven**. These functions provide basic building blocks for providing a micro service based architecture.

Pubic Cloud vendors have created an Infrastructure to provide Functions as a Service with Fully managed infrastructure and named them as Serverless Functions. All the major cloud vendors today are providing Serverless functions for use. Some of the widely used Serverless functions we know today are AWS Lambda Functions, Microsoft Azure Functions, Google Cloud Functions and IBM functions.

On the other hand, Open source platforms are emerging in the FAAS space that provides Function portability and more flexibility in terms of development .Few of the promising ones are Kubeless, Fission, KNative , OpenFAAS. These can be deployed on Kubernetes, a popular container orchestration platform.

Enterprise who choose to take the Serverless path can pick today from these two options:**Vendor provided Serverless Cloud platform** or **Vendorless Open Source Serverless Platform** based on their specific need.


Web and Mobile Customer Experience Layer
AWS Lambda
Azure Functions
Google Functions
IBM Functions
Vendor specific Serverless functions on Public Cloud
Customer Experience Layer (CMS, Angular JS , Xamarin, iOS, Android , Cordova, React , ReactNative)
Serverless Business Logic Layer
Web and Mobile Customer Experience Layer
OpenFAAS
KNative
Fission
Kubeless
Open Source Serverless functions on Kubernetes , Docker and Container Registry

Serverless -Vendor Specific and Open Source Options

Serverless: The Future of Digital Landscape

In a 'Digital First' world, the Customers, Partners and Internal users of an Enterprise need all services via digital touchpoints to self-serve themselves, anytime, anywhere. The demand for faster development and quick deployment of features is rapidly growing. The applications need to be scalable and agile. By combining Serverless functions and other cloud services, Enterprises can build the most powerful web and mobile applications that can scale up and down based on the demand. Key areas that we may benefit from Serverless are in the space of Customer journeys involving marketing content and e-commerce touchpoints.

Netflix (Refer Credits) is one of the biggest Industry leaders and trendsetters in using Innovative and new technologies. Netflix has moved its popular Video Content Streaming site from On Premise Data Centres to a completely Cloud Native approach. It is intensely utilizing Serverless functions to accomplish Media Content Sorting, deploying, publishing, approval workflows, monitoring using Lambda and other connected AWS services. Without Serverless, it was difficult to achieve such a highly scalable and robust platform to server millions of Subscribers.

ShutOut (Refer Credits) , which is a Stratup and offers a SAAS platform for improving Customer Engagements, Communication and Loyalty Management platform have recently transformed their platform using Serverless architecture. This provided them huge benefits to deliver innovative ideas faster than ever and at scale with reduced costs.

Some of the other major players in the market who have embraced Serverless and reaped great benefits are Coca Cola, BBC and AIRBNB.

Coca Cola (Refer Credits) reduced its Operational Costs by going Serverless.

BBC (Refer Credits) has created a Serverless Mobile platform based on AWS Lambda and API gateway.

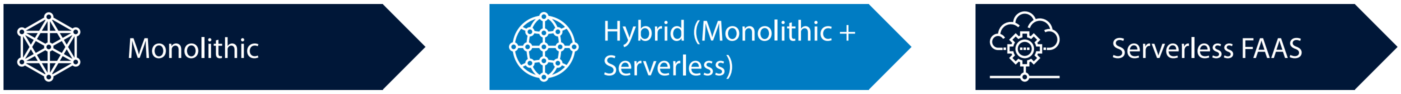
AirBNB (Refer Credits) has designed its own Alert Streaming framework using Serverless architecture.

With all these examples of overcoming adversity it is unequivocal that the fate of the Digital space is Serverless and an ever-increasing number of organizations ought to acknowledge this change and consolidate Serverless as an essential piece of their Digital Strategy.

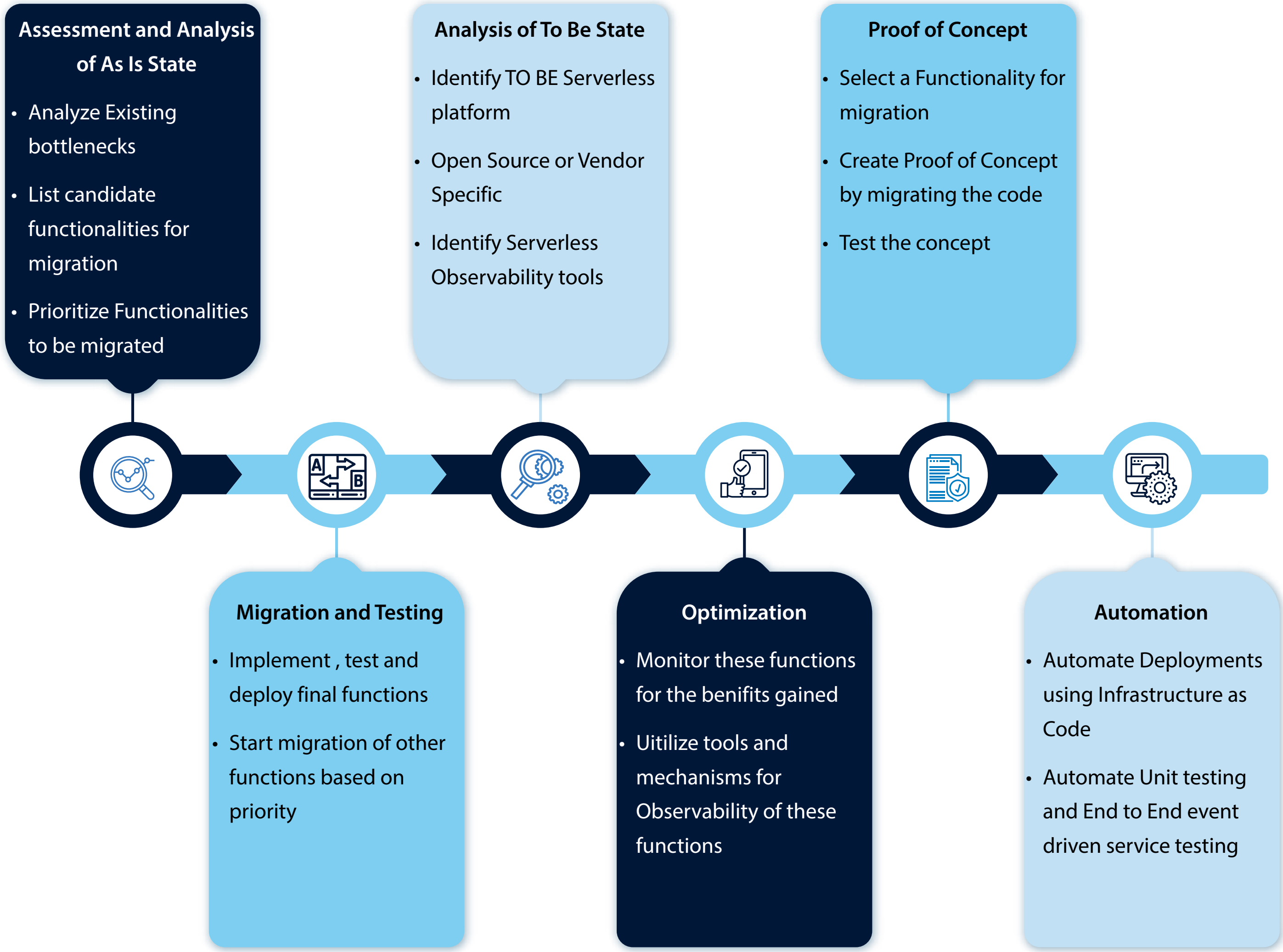
Recommended Approach to Modernize an Application using Serverless

An enterprise on an existing platform that is on a monolithic architecture and confronting issues regarding Granular scaling of the application, Higher Operational administrative expense and lower flexibility. The recommendation is to start thinking in the direction of modernizing its landscape using Serverless and event driven functions.

The critical approach is to isolate modules that encompasses the core of the business, yet can be modularized and hosted as a set of functions that can scale by itself. Rather than adopting a big bang migration strategy, the recommendation is to go for a simple stepwise, iterative methodology in which bottlenecks in the present functionalities are identified .Selected code is then migrated to the Serverless architecture. In this move from Monolithic to completely Serverless FAAS, there will be a phase where the application will work in a hybrid mode that despite everything will have numerous advantages over completely monolithic system.



Recommended Migration Approach



Assessment and Analysis of As Is State

- Analyze Existing bottlenecks
- List candidate functionalities for migration
- Prioritize Functionalities to be migrated

Analysis of To Be State

- Identify TO BE Serverless platform
- Open Source or Vendor Specific
- Identify Serverless Observability tools

Proof of Concept

- Select a Functionality for migration
- Create Proof of Concept by migrating the code
- Test the concept

Migration and Testing

- Implement , test and deploy final functions
- Start migration of other functions based on priority

Optimization

- Monitor these functions for the benifits gained
- Utilize tools and mechanisms for Observability of these functions

Automation

- Automate Deployments using Infrastructure as Code
- Automate Unit testing and End to End event driven service testing

Serverless Architecture for Core Digital Platforms

Customer Experience Platform

The Customer Experience layers for the digital applications is moving from heavy weight applications / portals we have seen in the past to lean user interfaces or lean portals. These are built using frameworks like Angular, React, HTML5, JS, React Native etc. The present need is to keep the logic at user interface very light so that the developers could focus on only the interface design and create best in class user experience without bothering much about managing and deploying the business logic layer. All the business logic need to be moved to a separate layer that could be reused across channels, managed and scaled independently of the presentation layer. Lightweight User Interface can consume backend functions and business logic using APIs exposed via API gateways. Most of the services available in the cloud now are HTTP enabled and the Customer experience layer can interact with these Cloud services via JWT tokens/ IAM roles or other similar mechanisms. This makes even easier to develop an entire application using a light weight Customer experience layer and a combination of Serverless functions in the backend and integration with other Cloud services for Authentication, Payment, Virus Scan, File Upload /Download (BAAS –Backend as a Service)

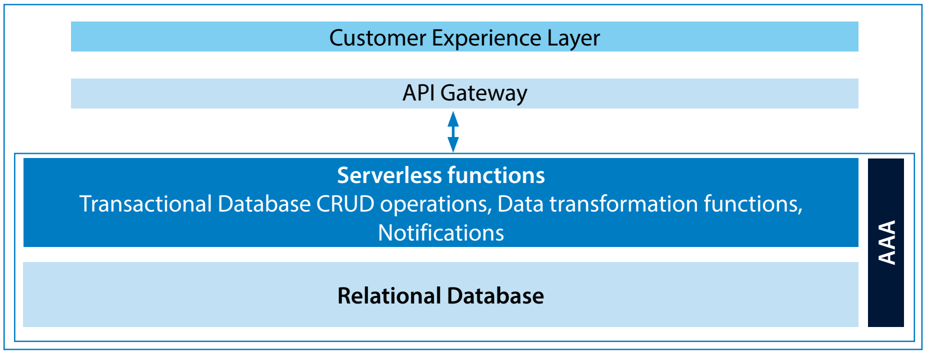
Bustle.com(Refer Credits), a news entertainment and fashion website has taken a major initiative to migrate all their Customer experience journeys on Mobile and Web from On Premise to Cloud based Serverless architecture. The new architecture is based on AWS lambda and other connected services.The company is gaining tangible benefits from the same.

A simplistic view of such a Serverless layer is shown below .This could be implemented using Vendors provided Serverless platforms or Vendorless Serverless platforms.

A typical implementation can use the following services from AWS Cloud platform –

- Lambda and Step functions
- RDS database
- Cognito for authentication and authorization.
- S3 for static content host

Serverless Customer Experience Platform


Customer Experience Layer
API Gateway
Serverless functions
Transactional Database CRUD operations, Data transformation functions, Notifications
Relational Database
AAA

Content Management Systems

The legacy Web CMS used by some Enterprise today are based on a Monolithic architecture and have their Presentation, Content Management and Content Delivery systems tightly coupled. These pose various challenges as stated below:

1. Channel- Centric
Earlier CMSs were built for websites and remain very channel centric. Because of this, Enterprises had to use different CMSs for multiple Channels.

2. No on-demand scaling
These CMSs have heavyweight ecosystems. They do not scale very well when needed.

3. Offer No or minimum agility
These systems do not offer agility in term of delivering content faster to the market and have rigid workflows.

4. Extensibility and Integration
Integration with other tools in the enterprise systems like PIM, Portals, Personalization etc. become difficult.

Because of these challenges, product vendors and companies are exploring various ways to implement lightweight, Micro functions based CMS systems that also scale on demand. Headless (API first), Serverless approach fits best here.

Many Product vendors are embracing this need and have some good offerings in this area like Contentful, Webiny and Cosmic JS. These provide Headless Serverless CMS systems.

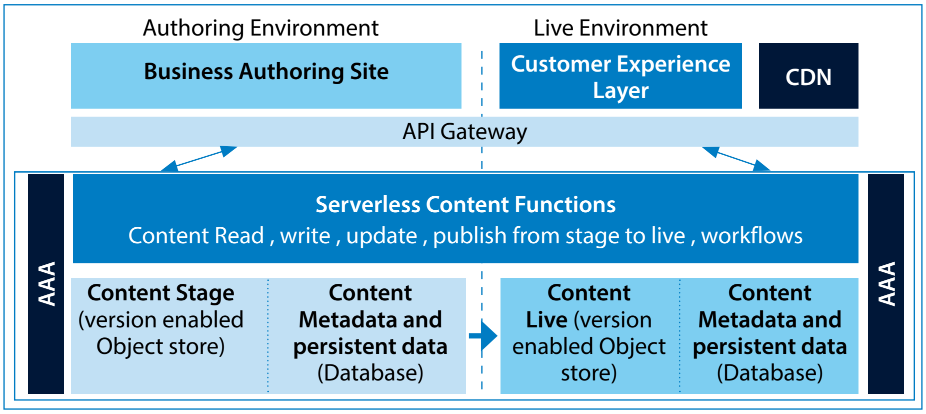
Headless systems provide API-first approach and provides a decoupled solution for Content Presentation and Management. Serverless take this to the next level and provide on demand scaling and lower Operational admiration costs.

Some of the CMS patterns to address this need are:

1. Large Legacy Enterprises
These Enterprise want to move away from monolithic Web CMS system can try modernizing their stack using

Headless Serverless CMS systems .They can try to pilot a green field project using the available Headless Serverless CMS systems and after reaping the said benefits and proving their Business Case, they can slowly start replacing other systems as well.

Headless - Serverless CMS Approach


Authoring Environment
Live Environment
Business Authoring Site
Customer Experience Layer
CDN
API Gateway
AAA
Serverless Content Functions
Content Read , write , update , publish from stage to live , workflows
AAA
Content Stage (version enabled Object store)
Content Metadata and persistent data (Database)
Content Live (version enabled Object store)
Content Metadata and persistent data (Database)

Altassian (Refer Credits) is a software tool provide company with Australian roots and Global reach.

Atlassian has recently leveraged Contentful to manage content for a growing list of external support services: help and technical documentation, FAQs, help articles, product documentation, in-product support, and resources for people considering buying Atlassian products.

Contentful is a SAAS solution that offers Headless Serverless platform. It handles spikes in traffic, manages scalability, flexibility to add third party integrations and expansion to Global markets.

2. Small and Medium businesses and Stratups
Small and Medium businesses, who want to start small initially, should consider build a basic Content Management System from scratch using public or open source Serverless options. They can also consider Open source CMS systems available in the markets that run on Serverless like Aerosol CMS and Webiny.

Such businesses can implement Static content hosting using object stores like S3 buckets and deliver them using Content Delivery Networks (CDN).More complex features like content authoring, inline editing, content workflows, themes / templates , User segmentations can be implemented incrementally as and when the business grows using a well thought design and Serverless backend.

3. Hybrid Approach
Companies currently using a Headless CMS approach but facing some challenges in terms of scalability can start modernizing some of their functions using Serverless approach.

E-Commerce Systems

E-Commerce sector is facing major business challenges today. The companies need to meet high customer satisfaction, lower wait times on site, and reduce the wastage of resources during downtimes, scale during peak hours and all this with reduced administration and operational cost. Few of the challenges faced by an e-commerce business are:

Need On demand scaling
These systems have fluctuating demands throughout the year and have a great increase in the demand during special occasions like Black Friday, New Year, and Christmas.

In a traditional monolithic approach, the Enterprise has to plan for the peak time capacity and use the required no. of instances. Even if they support auto scaling, individual and granular scaling of features is not supported.

Lower Operational Cost
Enterprises usually purchase the systems today from popular product vendors and have to pay overwhelming license cost .On top of the licenses cost the Administrative Operational cost and Administrative efforts of such monolithic traditional systems is another cause for concern.

Reduce Time to Market
Any new ideas in a monolithic application takes time for development and testing and hence, the overall time to market increases.

Some e-commerce patterns to address these bottlenecks:

1. Large Legacy Enterprises :
Large enterprises leveraging vendor based COTS products need to reduce Cost and increase scalability can consider modernizing their stack by going Serverless as explained earlier in this document.

MatHem (Ref. Credits), Sweden's largest online grocery store has recently redesigned their site using AWS Serverless functions. This helped them in achieving a highly scalable on demand web site .Also, lowered the time to market for new ideas and reduced overall operational cost.

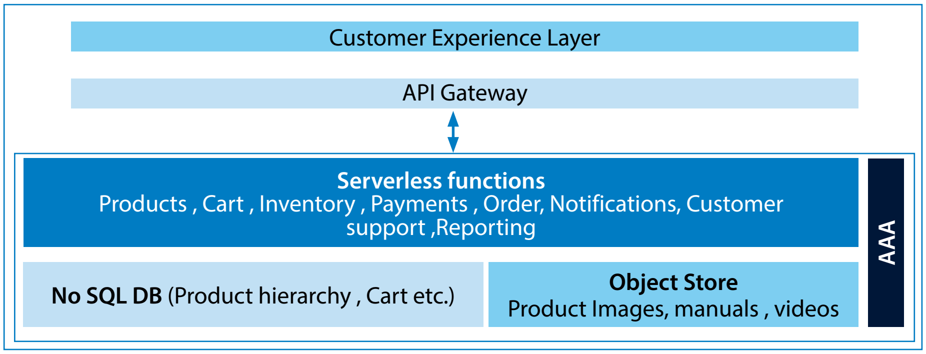
2. Small and Medium Businesses and Startups:
Smaller businesses and Startups, who cannot invest huge amounts in purchasing a COTS e-commerce product, can build up their own custom online shop platform easily using the Serverless approach.

3. Hybrid Approach
Vendorless Headless approach may have some gaps and hence can embrace a Serverless extensions against an extension on existing Vendorless platform itself.

An example use case in the Hybrid approach is building a Global Inventory view for Retail sector in near real time .This will help in implementing Omnichannel fulfilment capabilities.

The headless Serverless MVP can be easily achieved using a combination of the following:
Lambda Step functions, No SQL DB like Dynamo DB and S3 object store. AWS Cognito can provide Identity and Access Management.

Headless Serverless e-Commerce



Customer Experience Layer

API Gateway

Serverless functions
Products , Cart , Inventory , Payments , Order, Notifications, Customer support ,Reporting

No SQL DB (Product hierarchy , Cart etc.)

Object Store
Product Images, manuals , videos

AAA

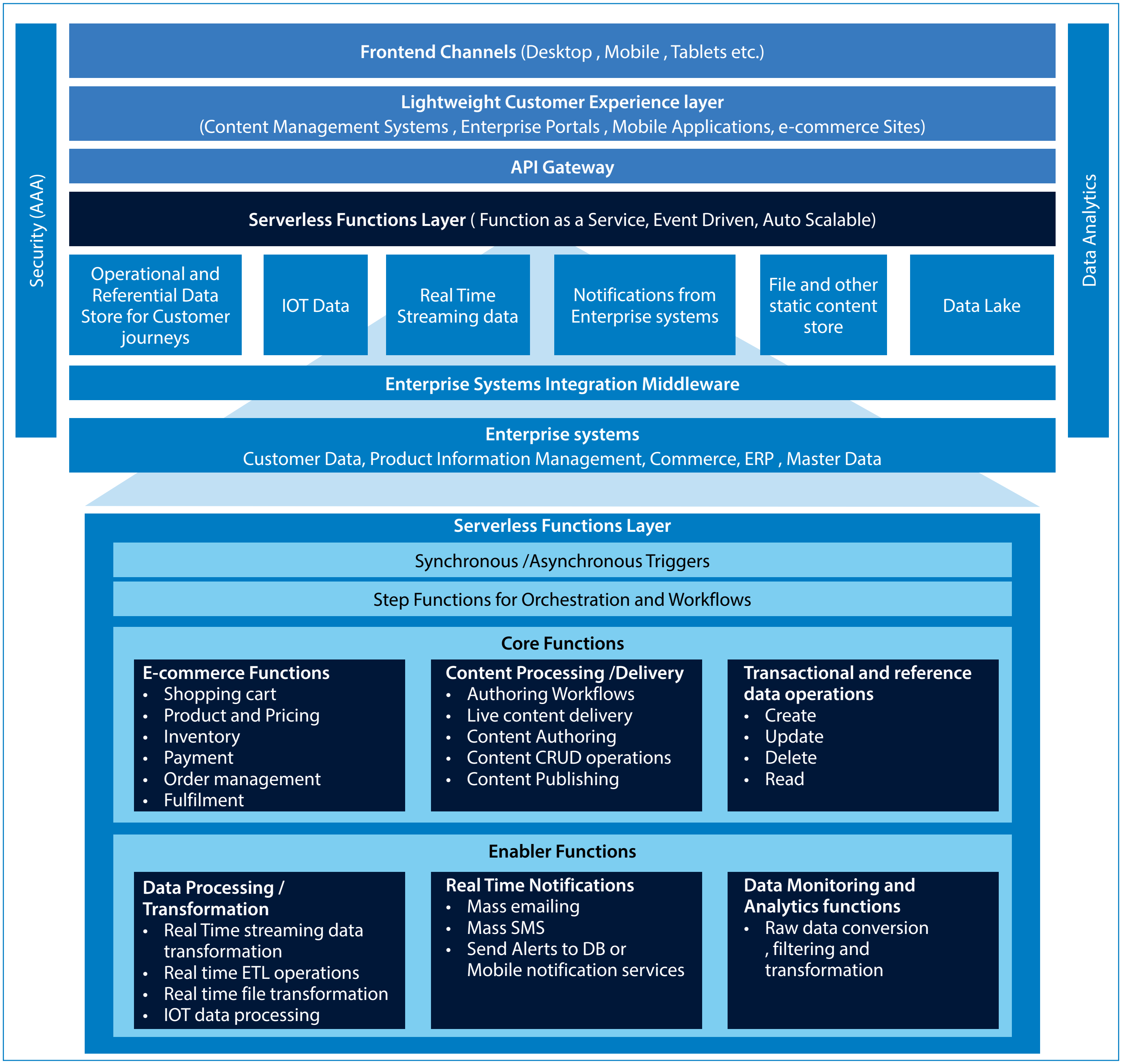
Reference Conceptual Architecture

A consolidated reference architecture that shows the use of Serverless Layer in an Enterprise Digital Landscape is illustrated below.

Serverless functions play a key role and help us in implementing the new Serverless function layer that implements business logic for applications. These Serverless functions act as micro functions and expose backend functionalities as micro services. Each functional unit can be developed, deployed and scaled separately.

The layer can be divided into the following.

- Function Invocation Layer (Invoked from API gateway or triggered based on an event from any other Cloud service like Lambda, Kinesis, SQS, SNS S3 etc.)
- Function Workflows and Orchestration (Lambda step functions)
- Core functions comprising of Content Management , e-commerce and Customer Experience Data base functions (Lambda functions with ECS tasks)
- Enabler functions comprising of Data transformation, processing functions and Notifications. (AWS Glue, Lambda, Kinesis, ECS etc.)


Security (AAA)
Data Analytics
Frontend Channels (Desktop , Mobile , Tablets etc.)
Lightweight Customer Experience layer
(Content Management Systems , Enterprise Portals , Mobile Applications, e-commerce Sites)
API Gateway
Serverless Functions Layer (Function as a Service, Event Driven, Auto Scalable)
Operational and Referential Data Store for Customer journeys
IOT Data
Real Time Streaming data
Notifications from Enterprise systems
File and other static content store
Data Lake
Enterprise Systems Integration Middleware
Enterprise systems
Customer Data, Product Information Management, Commerce, ERP , Master Data
Serverless Functions Layer
Synchronous /Asynchronous Triggers
Step Functions for Orchestration and Workflows
Core Functions
E-commerce Functions
Shopping cart
Product and Pricing
Inventory
Payment
Order management
Fulfilment
Content Processing /Delivery
Authoring Workflows
Live content delivery
Content Authoring
Content CRUD operations
Content Publishing
Transactional and reference data operations
Create
Update
Delete
Read
Enabler Functions
Data Processing / Transformation
Real Time streaming data transformation
Real time ETL operations
Real time file transformation
IOT data processing
Real Time Notifications
Mass emailing
Mass SMS
Send Alerts to DB or Mobile notification services
Data Monitoring and Analytics functions
Raw data conversion , filtering and transformation

Service Mapping for Serverless Layer and Supporting Functions

Note: The services are only indicative .Every Cloud offering has numerous services for similar functionality and should be evaluated based on the Enterprise need

		Public Cloud Vendors				**Open Source**
Layer in Reference Architecture	Services	AWS	Microsoft Azure	Google	IBM	Kubeless / Open FAAS
Security	Authentication	Cognito	Azure AD	Google Firebase	Identity Service	Flexibility to use any Open Source , Public cloud services
API Gateway	API Gateway	AWS API gateway	API Management	Google Cloud Endpoints	API Connect	Flexibility to use any Open Source , Public cloud services
Serverless Function Layer	**Serverless Functions Layer**	**AWS Lambda**	**Functions**	**Cloud Functions**	**Open Whisk**	**Kubeless /Open FAAS functions deployed on Docker Container**
Data Layer	Object Storage	S3	Blob storage	Cloud Storage	Object Storage	Flexibility to use any Open Source , Public cloud services
Data Layer	Relational data storage	RDS	Azure database	Cloud SQL	Data Services	Flexibility to use any Open Source , Public cloud services
Data Layer	Data Layer: Non Relational Data storage	Dynamo DB	Cosmos DB	Cloud Data Store	Cloudant	Flexibility to use any Open Source , Public cloud services
Data Layer	IOT data	AWS IOT	Azure IOT	Google IOT	IBM Watson IOT	Flexibility to use any Open Source , Public cloud services
Data Layer	Notifications	AWS SNS	Azure Notifications Hub	Cloud Messaging	IBM MQ	Flexibility to use any Open Source , Public cloud services
Big data and Analytics	Data Analytics	Redshift Kinesis Elastisearch	Synapse Analytics Stream Analytics	Google Analytics Cloud Data Flow	Smart Analytics	Flexibility to use any Open Source , Public cloud services, Kafka event streaming
Monitoring	Monitoring	Cloud Trail, Cloud Watch and Third party	Operational Insights	Cloud Console	Monitoring and Reporting	Flexibility to use any Open Source or other tools

Business Benefits of Proposed Serverless Architecture

The proposed architecture serves following benefits

- **Flexible and Adaptive**

Creating a Serverless functions layer provides flexibility to independently add, remove and update newer logic based on the current business requirements. Serverless functions provide capability to be invoked synchronously or asynchronously based on certain events. These can be exposed via an API gateway for consumption or can be invoked based on certain events such as Data change events in DB, File upload/modify events, Message Creation/Deletion Event in Message Brokers

- **Agile and faster development**

Functions are smaller and can be developed faster, deployed independently and can be added any time based on new business needs.

- **Separation of Concerns**

Each layer depicted in the proposed architecture addresses a specific need .The Serverless Function layer focuses on data processing and providing Business logic for the presentation layer

- **Reusable Business Logic**

The business layer functions are reusable and can be consumed by all the User touchpoints /channels via APIs.

- **Highly scalable and Available**

Serverless architectures can build powerful web applications that can auto scale up and down based on the demand and run in high available cloud environments. Scalability, backups are taken care of by the Cloud provider. Each function is independently scalable. For example Lambda functions can auto scale. Based on the no. of requests received by each lambda function it can increase the no. of instances running or reduce the instances.

- **Low Operational Administrative efforts and Cost.**

Serverless functions do not need server software or hardware management by the enterprise or developer. It provides a Pay per Use model and does not incur any cost when the function is not being used. The server software /hardware management is taken care of by the Cloud Vendor hence reducing Operational Administrative efforts and cost.

- **Flexibility to choose development language**

Serverless functions support multiple languages and provide a choice of language for implementing each use case .Some of the supported language provided by Lambda functions are Java, Go, PowerShell, Node.js, C#, Python, and Ruby code, and provides a Runtime API which allows you to use any additional programming languages to author your functions.

- **Connected Cloud Services and Event Driven**

Serverless functions are very well connected to other Cloud services in the ecosystem. This enables the architects to tackle many complex scenarios that may be encountered in a complex digital landscape.

For example -Lambda functions can be triggered synchronously and asynchronously by multiple AWS services.

- **Integrated Security**

Serverless function providers have implemented the security using IAM policies at very granular level.

The following is controlled via Security policies

- The access and trigger to these functions
- Access to other services from these functions

- **Operate in Multi Cloud Environment**

Enterprises today want to utilize the best services from different cloud vendors. They would want to utilize AWS API with Lambda and Google IOT.

In these scenarios, they have flexibility to deploy a mix and match of lambda functions in multi cloud environment.

Complex Usecase Implementation in Serverless World

Consider the applicability of Serverless architecture with an intricate real life example.

A Manufacturing Enterprise that builds and sells agricultural devices and equipment including connected Machines i.e. tractors and implements. The company is embarking on a Digitalization journey and wants to provide Digital Touchpoints to its consumers via a Self-Care Web and Mobile Channel.

The following key Customer Experience journeys for the end customers can be envisaged which can be achieved using a Serverless architecture.

Customer Journey	Serverless Architecture Applicability
Use Case 1: Machine Management Viewing machine fleet and Monitoring the machine telemetry data by processing real time IOT data	Real time IOT data processing and storing in Data store for display in the Customer Self-care site. This data could be anything from current machine location, machine hours , fuel level etc.
Use Case 2 : Field and Task data Management and Processing Processing of agricultural data related to Fields and Tasks to be executed on fields. View Access Field and Task data from Data Store	Converting and storing data received by Machines (recorded fields and completed tasks)into data format that can be stored in data store of the Customer self-care site and also send the updated data(updated fields and created tasks) from data store to Machines CRUD operations on the data store that have to be exposed to the portal. The data includes Field data, task data and Machine telemetry or any other reference data.
Use Case 3 : Using data analytics to improve portal experience e-commerce journeys to buy Machines and Accessories	Use data analytics to improve customer experience
Use Case 4 : Receive Notifications	Send Machine specific notifications or any other alerts relevant for the user.
Use Case 5 : Shop Online for Machine Parts	An after sales online shop that offers purchase of tractor parts.

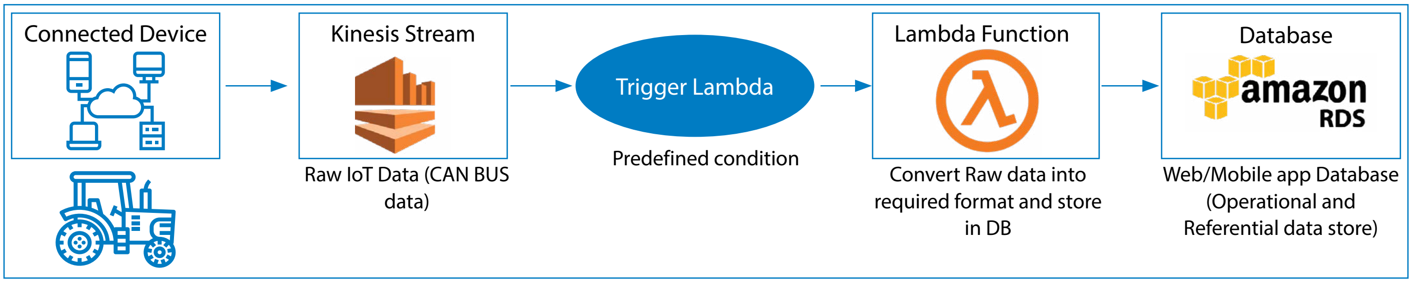
Sample implementation of the above journeys -

1. Use Case 1: Machine Telemetry data Stream processing (Building a Real time 'Extract Transform Load' pipeline)

With the increase in connected devices and the need to monitor these in Real time in applications, we believe that IoT data storage and processing is an important feature where Serverless functions may play a key role.

Machine / Connected device telemetry data received via a CAN BUS and be piped into Kinesis Streams and can then be processed using lambda. This data can be analysed, filtered, transformed and stored in Data Lakes and Application databases for further analysis and real time display in dashboards. The telemetry data example are historical machine track, Machine Hours or Machine Fuel value. Alerts and notifications can be generated from this data to inform the user about specific actions to be taken on the vehicle/ equipment. Also, auto actions can be taken by ordering a new equipment or part, which needs urgent replacement.

Real Time Telemetry Processing


Connected Device
Kinesis Stream
Trigger Lambda
Lambda Function
Database
amazon RDS
Raw IoT Data (CAN BUS data)
Predefined condition
Convert Raw data into required format and store in DB
Web/Mobile app Database (Operational and Referential data store)

Use Case Fig: 1

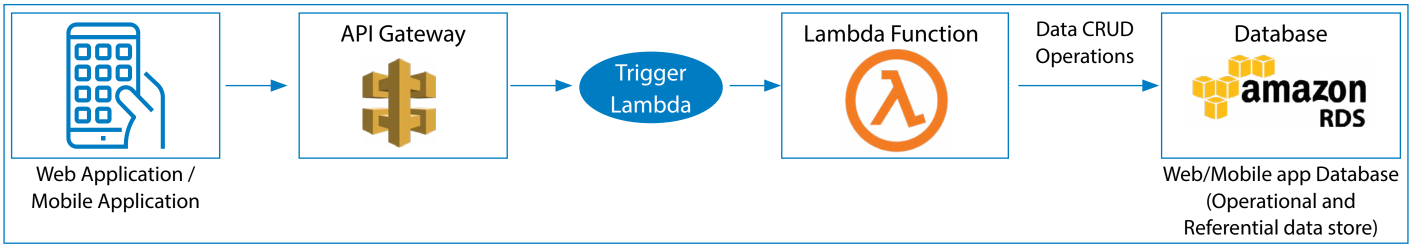
2. Use Case 2: Field and Task Data Management and Processing

a. Database Operations

Storage of Operational and Referential data is a key use case in any application. Serverless functions can help exposing this data in the form of micro functions to provide all CRUD operations.

In our sample, business scenario Operational and Referential data store is used to store processed Field, Task and Machine Telemetry data.

Database Operations



Use Case Fig: 2

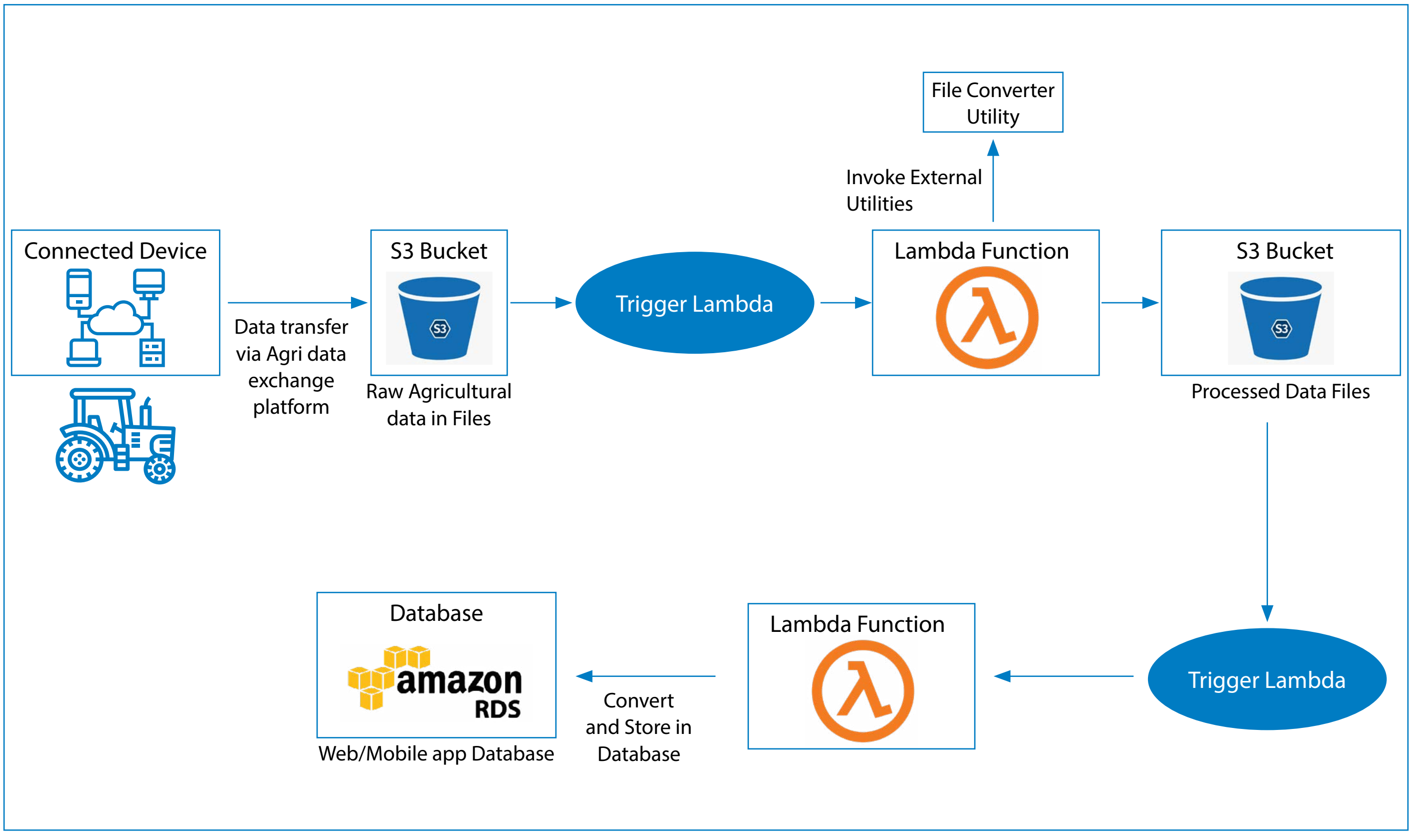
2. Use Case 2: Field and Task Data Management and Processing

a. Real Time File transformation functions

Raw data obtained from agricultural data exchange platforms could be placed in S3 buckets. Every write, update in S3 auto invokes a Lambda function.

These lambda functions can call external converters that help to convert the agricultural data files to specified format and vice versa. The converted files can be stored in Processed files S3 buckets.

Real Time IOT data and File Processing



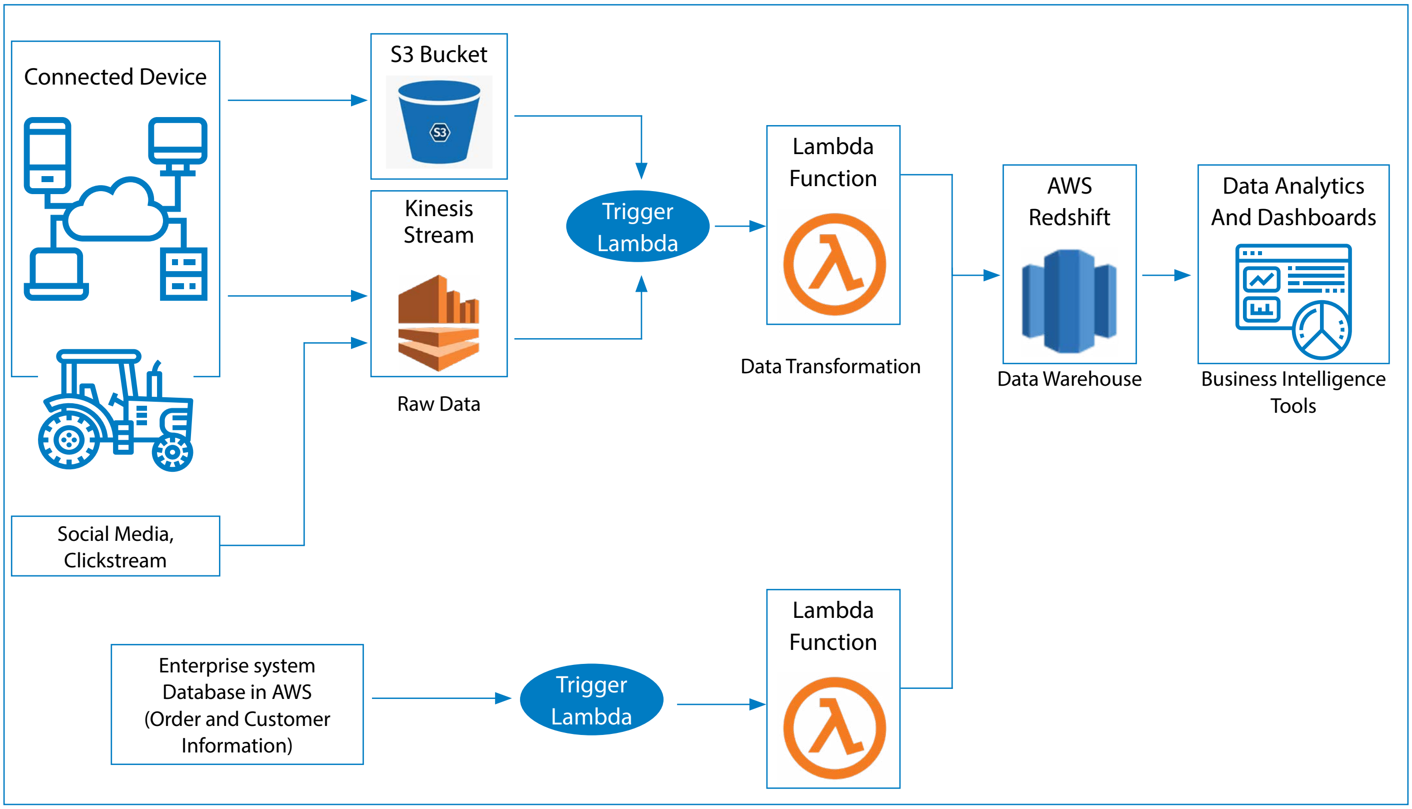
Use Case Fig: 3

3. Use case 3 : Data Analytics

Data Analytics is a must have functionality for any Enterprise today. Analytics help the Enterprise improve Operational efficiency, drive new revenue and gain competitive advantages over business competitors. Various forms of data is analysed today by any enterprise to gain important insights about their customer and products. This includes streaming data coming from Social Media, Telemetry data from Connected Devices and Vehicles, Website clickstream ,Customer Orders, Product Usage data . This data needs to be extracted from their source systems, transformed, sorted and loaded in Data Lakes. The Analysis of data and use of predictive modeling tools help companies to understand their customers , improve user experience of sites, do targeted promotions and run marketing campaigns , avoiding equipment failures , servicing the vehicles and improve their processes.

Serverless lambda functions help here as they can be easily triggered by Streaming data and S3 buckets .Lambda functions can be used easily to process this data , transform and load into data lakes. Business Intelligence tools like AWS Quick Sight , Tableau are used to analyse this data and present the results to Business users in Interactive Dashboards.

Data and Analytics



Use Case Fig: 4

4. Use Case :Notifications to Web and Mobile UI

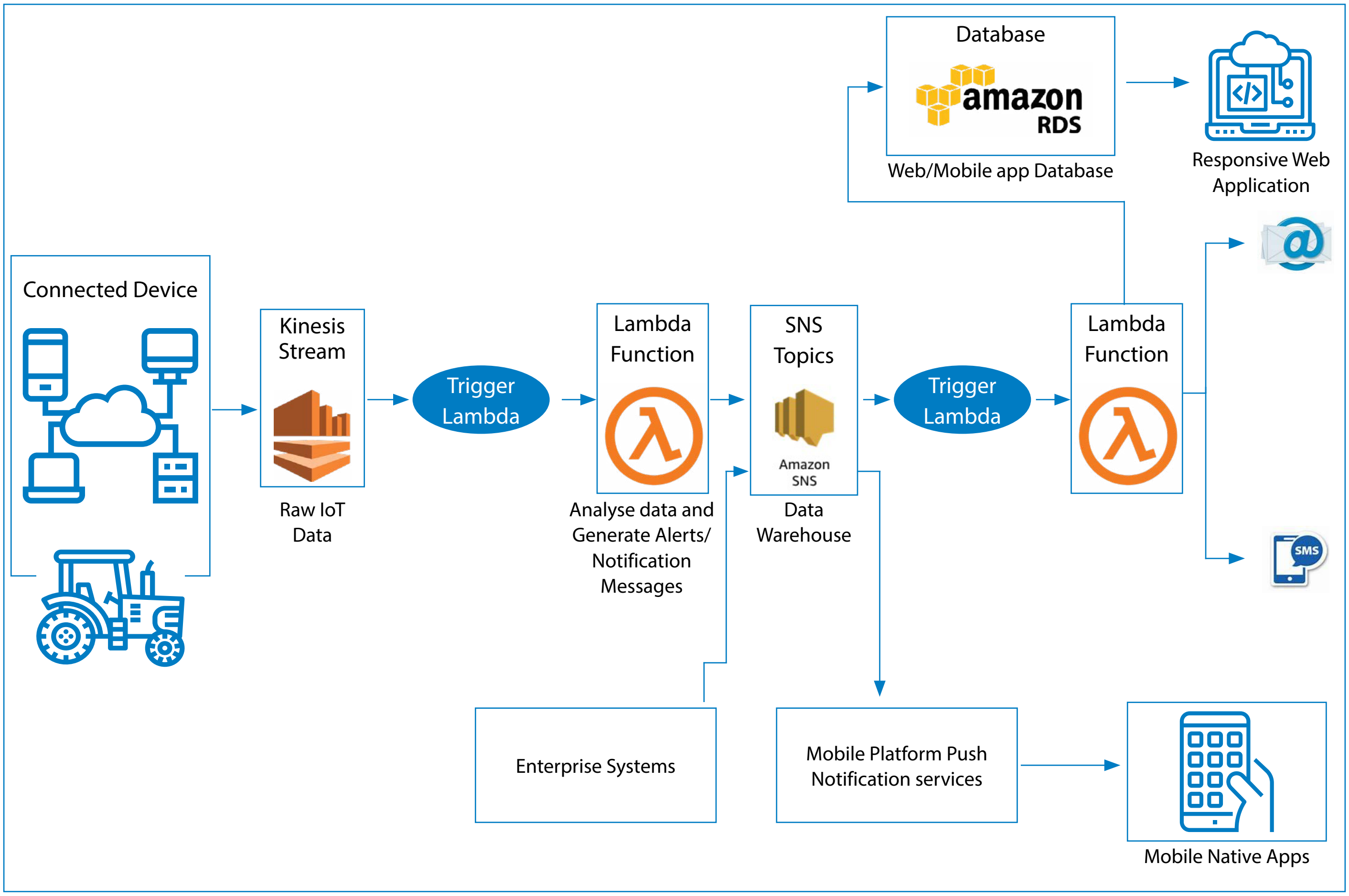
Push notifications, e-mails and SMS are effective tools to keep customers engaged with the applications and inform them or warn them about things that need their attention.

Lambda functions could be used to trigger such notifications. Example of such a notification is the Expiry of Vehicle warranty, Low Fuel level in Vehicle, Order dispatched, Order Delivered , New updates, Service Contact Expired etc.

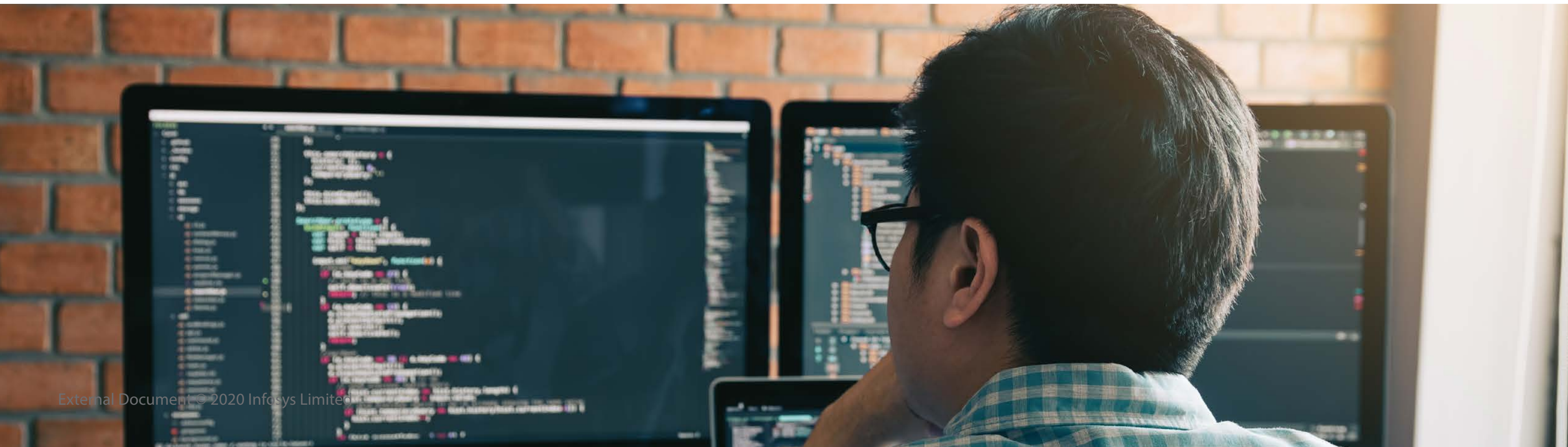
Data coming from IoT data streams or certain events in other enterprise systems like Order Creation, Order dispatched, Task Completion, Task Assigned can trigger lambda functions .Lambda functions can filter the required events and create required Notifications messages in AWS SNS service.

From SNS these messages can be pushed to Mobile via platform specific notification services like Apple Push Notifications and Google Push Notifications and to web application via inserting in application Database. Another set of lambda functions can be leveraged to provide some post processing on the messages like applying templates for eMail and SMS.

Mobile and Web Application Notifications



Use Case Fig: 5

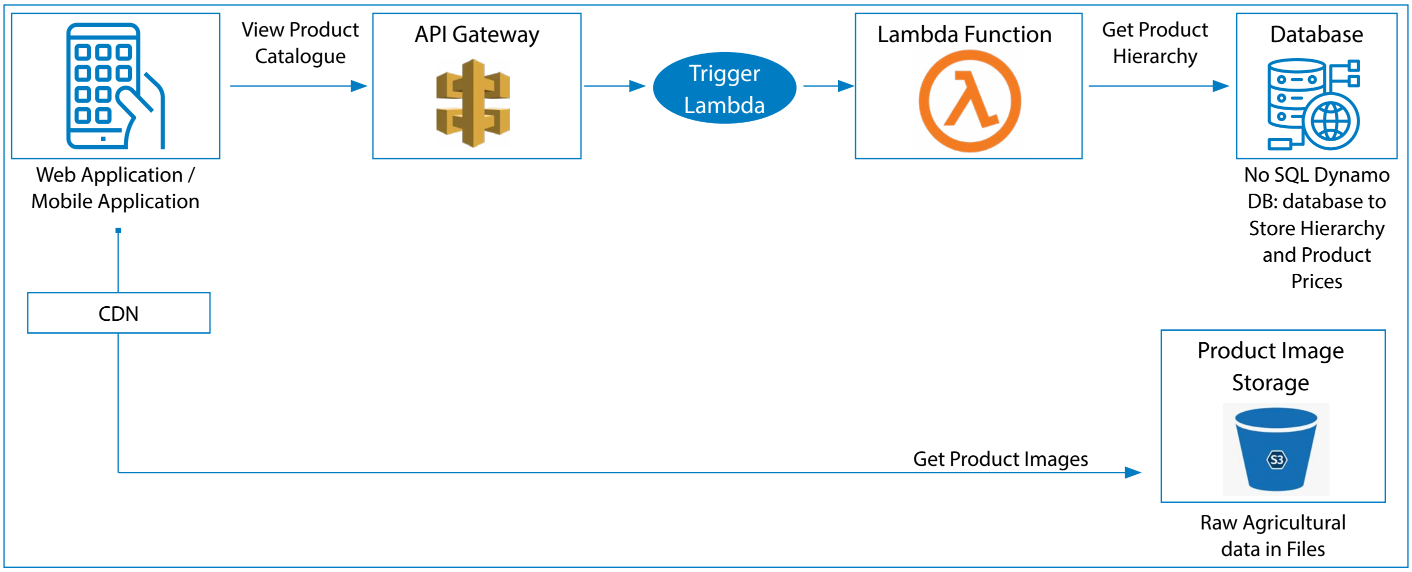


5. Use Case : On line Shop for purchasing Machine Parts

An engaging e-Commerce site is necessary for any Manufacturing company. This can be easily implemented using a combination of Object stores, No SQL , SQL databases and lambda functions.

A simple use case of 'Fetching the Product hierarchy' is provided below.

Online Shop



Use Case Fig: 2

Critical Challenges faced in Serverless Approach

Like any other technology, Serverless architecture also comes with certain constraints and pose challenges .There are workarounds and workable solutions to get over these limitations. The Serverless functions provided by public Cloud vendors come with a few constraints like function execution time limitations, Data payload size limitation and cold starts. However, these limitations can be easily overcome by various means and combining Lambda with other services.

As an example AWS Lambda functions currently have an execution time limitation of around 15 minutes. This limitation can be easily overcome by leveraging Step functions to break down the processing logic into multiple steps and combining with ECS tasks. Lambda will work as an invocation layer and the actual heavy workload could be handled by the ECS tasks that will be invoked only when required.

One of another such major challenge is Monitoring of such a highly distributed, Even Driven Serverless system. Currently AWS lambda can be monitored and debugged only via AWS Cloud Watch console logs. It is difficult to work with Cloud Watch when data volumes is huge and debugging becomes a difficult task .However, some of the new emerging third party tools like Dashbird , Epsagon and IOPipe provide much advanced Serverless FAAS Observability and Troubleshooting capabilities and should be considered while going Serverless .

Open Source Serverless FAAS platforms offers an alternative approach to Vendor provide Serverless FAAS platforms. These platforms provide more flexibility for Serverless implementations .These can be deployed on Docker Swarm or Kubernetes. Open Source Serverless offers a vendor neutral approach, are flexible to cover more scenarios without constraints and offer function portability across cloud platforms. This allows deployment on premise, on laptop or any cloud of choice or multi cloud environment. The projects where the data size is more and execution times are lengthy, Open Source fits better.

Enterprises already using Kubernetes(either On premise or On public cloud vendor managed service) to deploy their existing micro services can easily migrate to any of the Open source Serverless platforms like Kubeless, Open FAAS, KNative and Fission. This approach will reuse the underlying Kubernetes service with micro functions deployed over it.

Conclusion

Proposed architecture of using Serverless functions for handling and managing business logic can play a very vital role in implementing Digital Customer Experience journeys. Core Enterprise journeys, Content Management and e-Commerce scenarios could easily gain advantage using our Serverless Reference Architecture. This architecture focuses on resolving the major enterprise need today to provide highly agile, lightweight and scalable environment for always changing business demands. On demand, granular scaling is one of the key areas that it addresses which is missing today in the PAAS platforms.

Enterprises could focus on providing the actual functionalities and a world class Customer experience leaving the worry of Operational Administration to the Cloud vendors providing the Serverless function capability. There are some challenges observed in Serverless approach. However, with the developing advances in Serverless technology, emerging Open Source platforms and third party tools it is possible to beat these minor hiccups and go Serverless. The benefits of Serverless far outweigh the small drawbacks that have been observed.

The eventual future of Digital platforms is 'Serverless First Architecture' and as we see, a large no. of the big players in the market have already embraced and executed this change to gain huge competitive advantages.

Any big Enterprise should consider modernizing their existing stack using Serverless approach and incorporate Serverless as one of the Key design goals in any of their forthcoming transformation programs and green field projects. Serverless needs to be integrated as a core part of an Enterprise Digital Strategy.

Serverless offers great advantages for Startups. Stratups should think of 'Serverless First Approach' as their key considerations in designing. This will help them to launch large-scale innovative ideas and robust features as per the current market needs in no time, with reduced workforce and reduced Cost.

A good Digital partner can help realize the Serverless Dream of an Enterprise and help them in their overall digital stack modernization venture.

About the Author

Archa Shukla

Senior Technology Architect, Infosys Digital

Archa Shukla has more than 15 years of experience in Delivering Complex Digital transformation programs across various industry domains Energy, Telecom, Manufacturing, Retail, Healthcare and Finance. She has led and delivered big programs leveraging public Cloud services and Serverless architecture that has integrations with real time IOT data.

References

https://aws.amazon.com/lambda/?nc2=h_m1

https://docs.aws.amazon.com/lambda/latest/dg/lambda-services.html

https://aeqdigital.com/article/why-you-should-use-a-serverless-microservices-architecture/

https://www.bizety.com/2018/01/09/serverless-comparison-aws-lambda-vs-microsoft-azure-functions-vs-google-cloud-functions/

https://www.stratoscale.com/blog/cloud/aws-lambda-features-limitations-practical-examples/

https://techbeacon.com/app-dev-testing/event-driven-computing-best-practice-microservice-architecture

https://searchbusinessanalytics.techtarget.com/feature/How-big-data-analytics-tools-can-help-your-organization

https://codeburst.io/how-to-build-a-serverless-e-commerce-portal-48cdb6d49474

https://www.simform.com/ecommerce-app-using-serverless/

https://nickmchardy.com/2018/03/building-a-serverless-cms-using-aws-and-hugo.html

https://agileengine.com/serverless-architecture/

https://dzone.com/articles/serverless-retail

https://www.cosmicjs.com/

https://www.contentful.com/

Credits

1. MatHem case study : https://aws.amazon.com/solutions/case-studies/mathem/
2. ShutOut case study : https://serverless.com/blog/why-we-switched-from-docker-to-serverless/
3. CocaCola, BBC, AirBnb and Bustle.com Case Studies

https://northstack.com/why-companies-are-moving-to-serverless-cloud-technology/

https://aws.amazon.com/

https://dzone.com/

4. Example Use cases : https://aws.amazon.com/
5. Altassian case study : https://www.contentful.com/case-studies/atlassian/
6. Netflix case study : https://aws.amazon.com/solutions/case-studies/netflix-case-study/

https://dashbird.io/blog/serverless-case-study-netflix/

7. https://www.cosmicjs.com/
8. https://www.contentful.com/
9. https://www.webiny.com/

For more information, contact askus@infosys.com

Infosys®
Navigate your next

Infosys.com | NYSE: INFY

Stay Connected SlideShare