HLD : Create My Infrastructure (Self assisted mode)
(Open Capability Lab)
Version : 0.1 (In-progress)
Author : Rajesh Verma

Rajesh Verma

—- Confidential and not to be shared —-

1. High Level Design for CMI (Create My Infrastructure) 5
2. Issues & Challenges 5
3. Mitigation & way forward 5
4. Benefits 6
5. Overall strategy 7
6. Architecture & Design : CMI 9
6.1. Functional Architecture : CMI 9
6.1.1. CMI : Functional Components List & Description 10
6.2. System Architecture : CMI 13
6.2.1. CMI : System Architecture Building Blocks (List & Description) 14
7. CMI : Configuration data 17
7.1. Applicable Environments 17
7.2. Applicable Deployment Platforms 17
7.3. Applicable E2E Architecture Templates 18
7.4. Applicable Architecture Building Block (ABB) on AWS Cloud Platform 18
7.5. Applicable Architecture Building Block (ABB) on GCP Cloud Platform 19
7.6. Applicable Architecture Building Block (ABB) on Azure Cloud Platform 21
7.7. Applicable Architecture Building Block (ABB) @ On-premises 23
8. CMI : Example E2E Architecture & ABB templates 26
8.1. List of [On-premises-ABBs] 26
8.1.1. ABB : Database <MySql> @ On-premises 26
8.1.2. ABB : Database <Postgres> @ On-premises 26
8.1.3. ABB : Cache <Redis> @ On-premises 26
8.1.4. ABB : Cache @ On-premises 26
8.1.5. ABB : Observability @ On-premises 26
8.1.6. ABB : Load Balancer @ On-premises 26
8.2. List of [AWS-ABBs] 26
8.2.1. ABB : Database <MySql> @ AWS 26
8.2.2. ABB : Database <Postgres> @ AWS 26
8.2.3. ABB : Cache <Redis> @ AWS 26
8.2.4. ABB : Cache @ AWS 26
8.2.5. ABB : Observability @ AWS 26
8.2.6. ABB : Load Balancer @ AWS 26
8.3. List of [Azure-ABBs] 26
8.3.1. ABB : Database <MySql> @ Azure 26
8.3.2. ABB : Database <Postgres> @ Azure 26
8.3.3. ABB : Cache <Redis> @ Azure 26
8.3.4. ABB : Cache @ Azure 26
8.3.5. ABB : Observability @ Azure 26

Rajesh Verma

—- Confidential and not to be shared —-

8.3.6. ABB : Load Balancer @ Azure 26
8.4. List of [Azure-ABBs] 27
8.4.1. ABB : Database <MySql> @ Azure 27
8.4.2. ABB : Database <Postgres> @ Azure 27
8.4.3. ABB : Cache <Redis> @ Azure 27
8.4.4. ABB : Cache @ Azure 27
8.4.5. ABB : Observability @ Azure 27
8.4.6. ABB : Load Balancer @ Azure 27
9. CMI : ABB Cartridge preparation approach 28
10. Use-cases : CMI 29
11. Outcome System Architecture (for customer) 33
11.1. System Architecture (AWS Production) 33
11.2. System Architecture (AWS Non-Production) 34
11.3. System Architecture (GCP) 34
11.4. System Architecture (Azure) 34
11.5. System Architecture (Others e.g. Oracle, RedHat, etc.) 35
11.6. System Architecture Generic 35
12. Outcome : Architecture TCO (Roadmap) 36
13. Solution Approach : CMI 36
13.1. Manage Environments : Solution approach 36
13.2. Manage Architecture Building Block (ABB) : Solution approach 36
13.3. Manage Deployment Platforms : Solution approach 37
13.4. Manage Architecture Templates : Solution approach 38
14. Provision Infrastructure : Solution approach 39
14.1. Provision Infrastructure : User Interface (HA Mode) 39
14.2. Preview of provisioned infrastructure (HA Mode) 40
14.3. Provision Infrastructure : User Interface (Non-HA Mode) 41
14.4. Preview of provisioned infrastructure (Non-HA Mode) 42
14.5. Provision Infrastructure : High Level Design 42
15. De-provision ABB : Solution approach 43
15.1. De-provision ABB : User Interface 43
15.2. Deprovision Infrastructure : High Level Design 43
16. Provision ABB : Solution approach 44
16.1. Provision ABB : User Interface 44
16.2. Provision Infrastructure : High Level Design 44
17. Import Legacy Infrastructure : Solution approach 45
17.1. Import Legacy Infrastructure : User Interface 45
17.2. Import Legacy Infrastructure : High Level Design 45
18. Appendix 46
18.1. ABB Cartridges @ AWS 46
18.1.1. AWS Route 53 46

Rajesh Verma

—- Confidential and not to be shared —-

18.1.2. AWS CloudFront 46
18.1.3. AWS API-GW 47
18.1.4. Application Load balancer 47
18.1.5. Network Load balancer 47
18.1.6. AWS VPC Peering 47
18.1.7. AWS WAF 47
18.1.8. AWS IAM 47
18.1.9. AWS CloudWatch 47
18.1.10. AWS CloudTrail 47
18.1.11. AWS S3 47
18.1.12. AWS Cognito 47
18.1.13. AWS KMS 47
18.1.14. AWS Secret Manager 47
18.1.15. AWS SQS 48
18.1.16. AWS SNS 49
18.1.17. AWS SES 49
18.1.18. AWS EC 49
18.1.19. AWS EC2 49
18.1.20. AWS EKS 49
18.1.21. AWS RDS 49
18.2. ABB Cartridges @ GCP 49
18.3. ABB Cartridges @ Azure 49
18.4. ABB Cartridges @ On-premises 49
19. Appendix-2 50
19.1. Scenarios/Rules to define ABB Cartridges 50
19.2. Bigger Vision : Open Capability Platform (OCP <LoCoNoCo>) 51
19.3. Zoom-in image of CMUX 53
19.4. Zoom-in image of CMUX 54
19.5. Integration Architecture 55

Rajesh Verma

—- Confidential and not to be shared —-

Preface

This article I am writing to share an Idea and Initiative named CMI (Create My Infrastructure).

Earlier I wrote an article [[S1-A&D] How to automate software-architecture, using architecture-as-code (SAaC) approach], This article in the same sequence is any attempt to share two type of information

1. Objective & Overview of CMI Platform (Covered below in the article)
2. High level solution approach to build the CMI Platform (Please follow the Link)

Small note on HLD (Solution Approach)

In order to keep the article size smaller, not sharing the entire solution-approach here. But the original document (Google Docs) will be available for all, interested & enthusiastic friends can follow the (Link of HLD document). For a long time, I was thinking of building this open-platform and [ABB cartridge] for each technology component, is the atomic component on the [CMI platform]. [ABB cartridge] is very much similar to [Provisioning-cartridges-in-Telecom] e.g. SIM-Activation-Provisioning-Cartridge, Change-SIM-Provision-Cartridge etc. for Telco operators. Technology components in any system architecture, in our architecture world is called [Architecture Building Blocks (ABB) (article on same here)] and [ABB-Cartridge] is the independent utility which can be download by user and execute in their infrastructure manually, or CMI platform can be allowed to executed in user's infrastructure automatically.

1. High Level Design for CMI (Create My Infrastructure)

— Confidential and not to be shared —

Overall objective of this project is to help the DevOps team/individual for setting up infrastructure for any software project in the required environments e.g. Production & Non-prod environments.

2. Issues & Challenges

1) Software vendors, system integrators and business entities having their own IT applications.
2) These stakeholders do setup IT infrastructure to run the respective applications
3) DevOps / other responsible teams setup these infrastructure manually
    a) On-cloud : via Cloud admin console or cloud scripts (e.g. CFS, TFS)
    b) On-premises : using installation scripts provided by each tool.
4) **Concerns & Observation**
    a) Step "**Infrastructure setup**" is usually performed in an unorganized & ad-hoc manner.
    b) Sufficient amount of time is taken to set-up the complete infrastructure.
    c) For each environment, similar steps are repeated with different configurations and the same amount of time is invested in those repeated tasks.
    d) Because of manual intervention, many-a-times human mistakes are committed.
5) All these concerns & observations are actually impacting efficiency, production and TCO (total cost of ownership).

3. Mitigation & way forward

1) A smart tool can solve all above problems efficiently
2) The tool can setup infrastructure for any environment (DIT, SIT, UAT, PT, Pre-Prod, Prod)
3) The tool can setup infrastructure on any platform (On-premises : Physical, Virtualize, Containerized | Cloud : AWS, GCP, Azure, etc.)
4) The tool can setup infrastructure as per pre-defined architecture templates.

Rajesh Verma

—- Confidential and not to be shared —-

4. Benefits

1. **Product & services as outcome of platform**
    a. Create my infrastructure (CMI) : This document is written to explains the solution blue-print of CMI.
    b. Create my business (CMB) : Blue-print for CMB shall be explained in separate document
    c. Create my user experience (CMUX) : Blue-print for CMUX shall be explained in separate document

2. **Implicit benefit from overall platform journey**
    - ❖ Increased productivity & enhanced efficiency
    - ❖ Time & Currency saving
    - ❖ Skilled Developer's community
    - ❖ Engaging Academic Pool for development
    - ❖ Promotion of Free & Open Source Software (FOSS) Community

3. **Many problems get resolved automatically**
    - ❖ Dynamic system architecture diagram generation
    - ❖ Dynamic functional architecture diagram generation
    - ❖ Dynamic integration architecture diagram generation
    - ❖ Water flow testing of entire platform
    - ❖ Platform readiness testing before GO-LIVE
    - ❖ Satellite view entire platform or component
    - ❖ Software-architecture-as-code (SaaC)
    - ❖ All of above fully or partially

Rajesh Verma

—- Confidential and not to be shared —-

5. Overall strategy

Detailed strategy for all below points shall be defined later, however high-level guideline will be following.

- **Open-source strategy**
  - I am great admirer of open source software, hence I wish to develop CMI product as "**Open Source Project**" initiative
  - Individual edition would be always part of "**Open Source Project**"
  - Enterprise edition, "**Open source**" strategy would be decided in future
  - **Note : In every situation, CMI would be very close to Open-Source principles**.

- **Customer segment**
  - CMI product shall be developed for "**Individual**" & "**Enterprise**" segments. Initially CMI would be launched for the developers community.
  - CMI product would be targeted for both kind of customers
    - who want to setup CMI for completely fresh infrastructure (i**.e. who don't have IT setup**)
    - who want to setup CMI for their existing infrastructure (**i.e. defining AS-IS legacy architecture into CMI**)
- **Product development strategy**
  - CMI product shall be developed in collaborated manner with help of developer community
  - CMI product's detailed blue-print shall be published to developer community

- **Rewards strategy**
  - Contributors shall be rewarded for their contributions
  - Reward criteria shall be defined in detail which will be based on many factors e.g. type of contribution, productivity etc.
  - Contributor's productivity shall be measured based on milestones achieved.
- **Revenue model**
  - "**Individual edition**" would be free of cost however donation request for "**Individual users**" will also be promoted
  - "**Enterprise edition**", revenue model would be decided later
    - Which could be either donation basis
    - Or yearly very nominal subscription fee e.g. 10 USD for each non-prod and 50 USD for each production environment
  - Note-1 : Difference between Individual & Enterprise edition can be Support license
    | OR |
  - Note-2 : Some advance features/utilities can be available only in Enterprise edition

Rajesh Verma

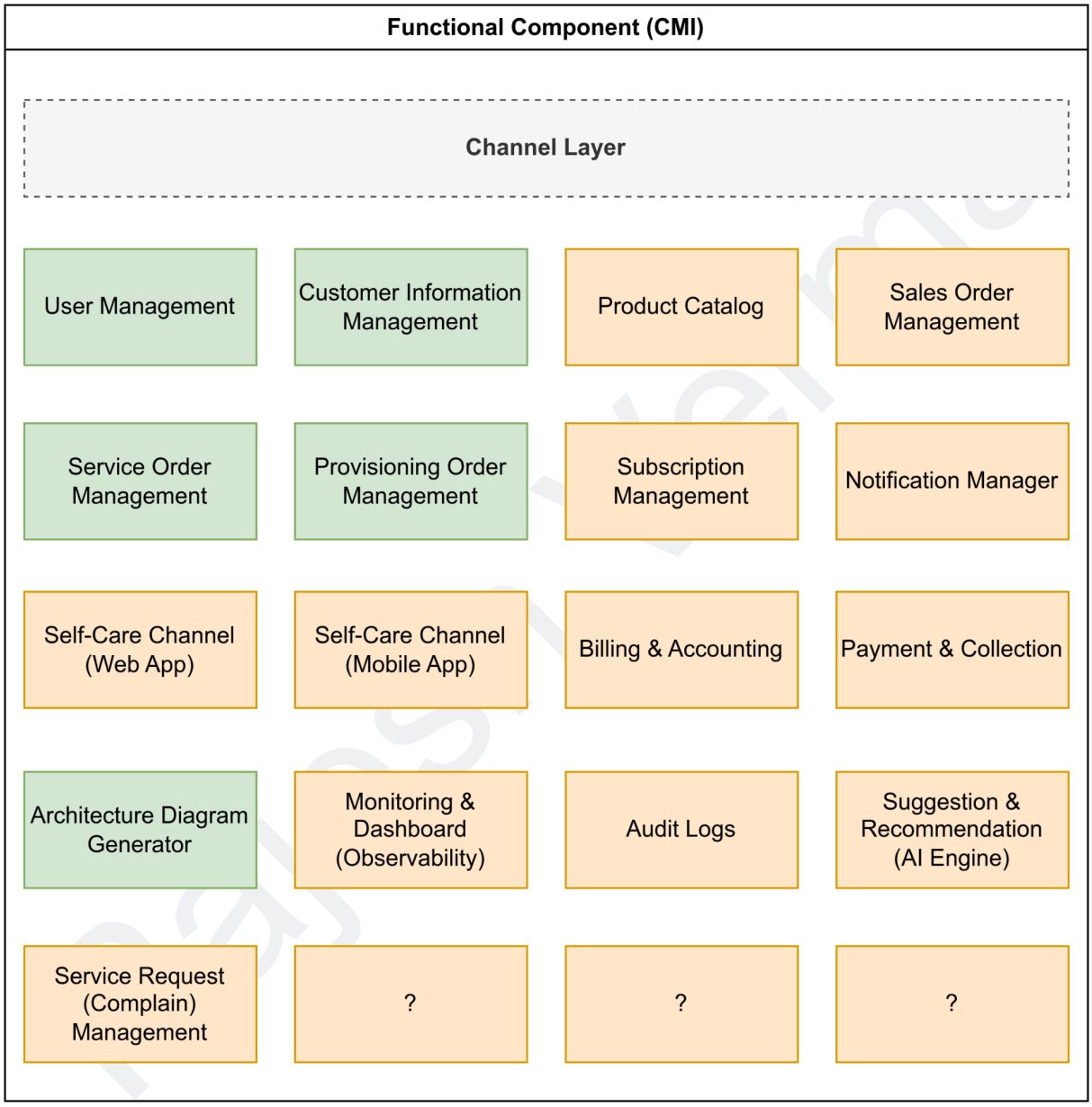
—— Confidential and not to be shared ——

Rajesh Verma

—- Confidential and not to be shared —-

6. Architecture & Design : CMI

6.1. Functional Architecture : CMI



—- Confidential and not to be shared —-

6.1.1. CMI : Functional Components List & Description

Seq	Functional name	Plan /Priority	Edition	ABE	Use-cases	Description
1	User management	MVP	Ind / Ent	User, Role	Manage user, password, OTP, role-mapping, authentication, authorization, etc.	CMI:UM will manage user related information & activities as explained in the UC column.
2	Customer Information Management	MVP	Ind / Ent	Customer, Customer-Account	Manage customer profile, contacts, addresses, customer life-cycle, account life-cycle etc.	CMI:CIM will manage customer related information & activities as explained in the UC column.
3	Product Catalog	Roadmap	Ind / Ent	Product, Product-Attributes, Offer	Manage product specification, attributes, prices, decomposition tasks & various rules, etc.	CMI:PC will manage product related information & activities as explained in the UC column.
4	Sales Order Management	Roadmap	Ind / Ent	Sales-Order, Sales-Order-Item	Manage sales order, sales-order life-cycle, sales-order-item life-cycle etc.	CMI:SOM will manage Sale-Order related information & activities as explained in the UC column.
5	Service (Technical) Order management	MVP	Ind / Ent	Service-Order, Service-Order-Task	Manage service order, service-order life-cycle, service-order-task life-cycle etc.	CMI:SeOM will manage Service-Order related information & activities as explained in the UC column.
6	Provisioning Order Management	MVP	Ind / Ent	Provisioning-Order, Provisioning-Order-Task	Manage provisioning order, provisioning-order life-cycle, provisioning-order-task life-cycle etc.	CMI:POM will manage Provisioning-Order related information & activities as explained in the UC column.
7	Subscription management	Roadmap	Ind / Ent	Customer-Subscription	Manage subscription, subscription life-cycle, renewal etc.	CMI:SM will manage Subscription related information & activities as explained in the UC column.
8	Notification manager	Roadmap	Ind / Ent	Notification-Template, Notification-	Send notification, receive notification, reminder notification, notification	CMI:NM will manage Notification related information & activities as explained in the UC column.

—- Confidential and not to be shared —-

				Transaction	life-cycle etc.	
9	Service Request (Complain) Management	Roadmap	Ind / Ent	Service-request, Followup,	Raise complain, Follow-up, SR life-cycle, Create SR, Create followup, Assign SR to group, Allocate SR to individual in group, Transfer SR ot other group or individual etc.	CMI:SR will manage Service-Request related information & activities as explained in the UC column.
10	Self-care Web-app	Roadmap	Ind / Ent			CMI:SCWA will be customer facing web application which shall enable customer perform query & provisioning activities
11	Self-care Mobile-app	Roadmap	Ind / Ent			CMI:SCMA will be customer facing mobile application which shall enable customer perform query & provisioning activities
12	Billing & Accounting	Roadmap	Ind / Ent	Charges, Adjustment, Invoice, Invoice Item	Prepare applicable charges, apply adjustment, perform billing, generate invoices, adjustment life-cycle, invoice life-cycle etc.	CMI:BA will manage customer billing for applicable subscription, waiver, late-fee, initiate termination request, initiate reminder notification etc.
13	Payment & Collection	Roadmap	Ind / Ent	Payment	Receive payment, payment allocation to invoices, payment life-cycle etc.	CMI:P&C will manage payment related information & activities as explained in the UC column.
14	Architecture Diagram Generator	MVP	Ind / Ent	Architecture Diagram	Generate system architecture, Generate functional architecture, Generate integration architecture, Generate deployment architecture	CMI:ADG will produce graphical representation of “System architecture” for selected environment & deployment platforms.
15	Monitoring & Dashboard (Observability)	MVP+	Ind / Ent			CMI:OMD will enable backend business & operation team to monitor various system metrics & business level KPIs 1) System metrics : CPU, Memory, Storage, IOPS 2) Business level KPIs : Number of requests, Number ABBs, Number of notifications, Number of invoice sent, Amount due,

Rajesh Verma

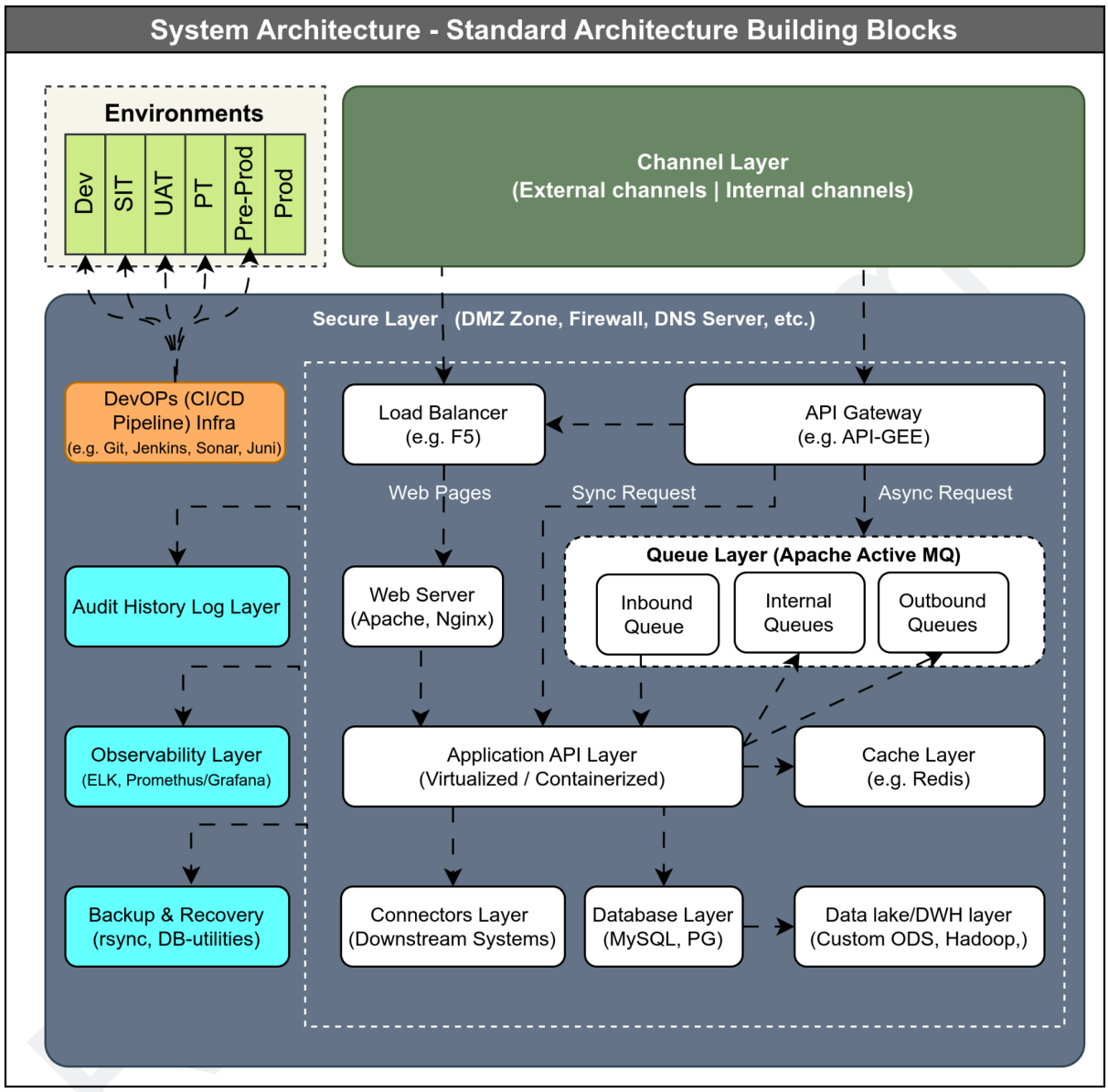
—- Confidential and not to be shared —-

						Payment collected, etc.
16	Audit Logs	MVP+	Ind / Ent	Activity Logs, Audit History DB	Create activity, Create history record, Generate activity log reports, Generate audit reports	Standard audit logs e.g. Activity logs, Data change (i.e. old & new values), etc.
17	Suggestion & Recommendation (AI Engine)	Roadmap	Ind / Ent			CMI:SnR will produce suggestions & recommendations for customers e.g. Resource utilization report and suggestion for optimization.

Rajesh Verma

—- Confidential and not to be shared —-

6.2. System Architecture : CMI



—- Confidential and not to be shared —-

6.2.1. CMI : System Architecture Building Blocks (List & Description)

Seq	ABB name	Plan /Priority	Description
1	Channel-layer		Channel layer will act as the first touchpoint towards the application or platform. This layer is also called omni-channel layer, usually companies want to have unified customer experience across all omni-channels. Mobile app, web-app, SMS, USSD, Social sites, partner's application and so on, there could be several channels for end-customers to interact with the application or platform.
2	Load-balancer		Load-balancer will act as a router & equal distributor of requests from the channel towards the next level architecture building blocks (i.e. web-server, service layer & queue layer).
3	Web-server		Web-server caters the web-page request i.e. based on web-page required content, web-server will pull data from backend, render the page with data and forward the rendered page to the user's machine browser. Example of web-servers are Apache HTTP server, NGINX Server, IIS server, etc.
4	Queue-layer (asynchronous request)		To process customer requests in asynchronous manner to optimize spike during peak hour and to also avoid long block threads e.g. JMS Queue, Apache Active MQ, Rabbit MQ, etc. Queues can be planned for Incoming traffic, Outgoing traffic and internal traffic itself. If the particular request processing requires complex computation at backend (e.g. multiple application integration, database tables fetch, business validation, transformation, request processing at 3rd party systems, publishing messages on various downstream systems etc.), it is recommended to opt for an asynchronous option e.g. spinning a virtual machine on cloud platform, buy a new mobile-connection with entertainment services like netflix, amazon-prime & Disney+ hotstart and so on.
5	Direct from Load balancer to Application API layer (synchronous request)		To process customer requests in a synchronous manner. If the request processing does not create a long-blocking thread (i.e. the particular request processing does not require multiple application integration, database tables fetch, business validation, transformation, etc.), it is recommended to opt for a synchronous option for faster response e.g. get my outstanding balance, get my current plan.

Rajesh Verma

—- Confidential and not to be shared —-

6	Cache-layer		Cache-layer is to keep frequently fetched information and will be used to enhance the overall application performance. Cache-layer is the optional component and should be considered if there is a need for faster response from application or platform.
7	Application API layer		Application API layer is the actual processing engine for any kind of request coming from the channel-layer. This works as a hosting layer for APIs (which are developed on the ecosystem) which will be consumed internally/externally and synchronously/asynchronously. ***API complexity ∝ Number of computation tasks in the*** *APIs can be simple or complex in term computatio complexity from simple-to-complex directly proportion number of computation tasks in the API.* To process customer requests in a synchronous manner to respond faster. APIs can be hosted in Apache Tomcat, RedHat JBoss server, Oracle WebLogic, Kubernetes Docker Cluster dependending on the need of a virtualized or containerized hosting.
8	Database-layer		Database-layer is persistence layer for the reference data & transaction data. Database-layer will be used for read & write operations of relevant data. Database-layer is one of most critical ABBs, depending upon the load of read & write operation and performance expectation, segregation of read & write operation should be done. Architecture of the database-layer itself plays an important role in the efficient-&-performant system design.
9	ODS (MIS Reporting) layer		Now-a-days none of the businesses can survive without a good MIS-reporting layer. Usually companies implement different kinds of DWH solutions depending on their budget and criticality. ODS is called an operational data store which can be considered as a basic alternate of complex DWH solutions. ODS is similar to database-layer only but this is dedicated for MIS reporting purposes. If there are a huge number of requests from channel-layer, it is not recommended to have a common database for your end-customer's request processing and internal MIS reports generation. ODS-layer is replicated from main-database-layer, there are various approaches of data-replication from main-database-layer to ODS-layer.
10	Audit History Log Layer		Audit-history-log-layer is a very important ABB in the ecosystem which enables the companies to visualize the historical changes on their business data. There could be various approaches to

Rajesh Verma

—- Confidential and not to be shared —-

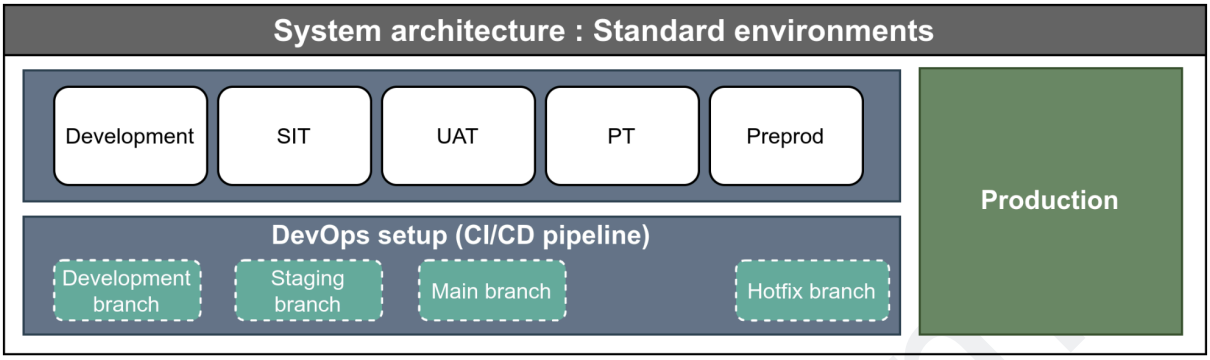
			implement audit-history e.g. write a custom-audit-layer which will hold the min information about data-changes and consolidate the sequence of changes whenever required. The simplest approach could be to have a history database-layer for each or relevant database entities with the same entity structure including history date-time stamp and consolidate data to show the historical change history whenever required.
11	Observability layer		Observability-layer is to keep centralized logging from every ABB and establish log-correlation for debugging purposes. Observability-layer will hold the logs in unified formation from each layer. Let's understand this with one example, suppose one request is sent from the end-user's browser which may pass-through from load-balancer, web-server, application-API-layer, queue-layer, cache-layer and others. Considering this example every ABB's log will be captured in the unified-log-format on the observability-layer (unique-correlation-id will be assigned for each request from channel), in case of issue with any request, observability-layer will leverage the companies to provide consolidated logs by correlation-id.
12	Backup & recovery		Backup-&-recovery-layer to restore the database snapshot in case of main-database crash due to any reason. Backup-&-recovery-layer will also define a policy to restore application from archive in case of any deployment issue.
13	DevOps-layer		DevOps is not the ABB which will participate in the request processing of requests coming from channel-layer, but this has become an important ABB which will automate the source-code version management, apply static code analysis, perform code-coverage & test-coverage checks and deployment on relevant environment depending upon %-coverage-pass policy.

Rajesh Verma

—- Confidential and not to be shared —-

7. CMI : Configuration data

7.1. Applicable Environments



Seq	Environment Name	Description
1	DIT	Development Integration Test
2	SIT	System Integration Test
3	UAT	User Acceptance Test
4	Pre-Prod	Pre production
5	PROD	Production

7.2. Applicable Deployment Platforms

Seq	Deployment Platform	Description
1	AWS Cloud	CMI will use AWS Cloud to deploy infrastructure
2	GCP Cloud	CMI will use GCP Cloud to deploy infrastructure
3	Azure Cloud	CMI will use Azure Cloud to deploy infrastructure
4	On-premises	CMI will use On-premises setup to deploy infrastructure
5	Other platform	
6	Other platform	

Rajesh Verma

—- Confidential and not to be shared —-

7.3. Applicable E2E Architecture Templates

Seq	Environment Name	Description
1	BAS-NON-PROD	Base architecture (BAS-NON-PROD) with mandatory ABBs
2	ADV-NON-PROD	Advance architecture (ADV-NON-PROD) with mandatory and all good-to-have ABBs
3	BAS-PROD	Base architecture (BAS-PROD) with mandatory ABBs
4	ADV-PROD	Advance architecture (ADV-PROD) with mandatory and all good-to-have ABBs

7.4. Applicable Architecture Building Block (ABB) on AWS Cloud Platform

Seq	AWS ABB Name	ABB Layer	Description
1	AWS Route 53	Routing Layer	Please refer AWS online document
2	AWS CloudFront	Routing Layer	Please refer AWS online document
3	AWS API-GW	Routing Layer	Please refer AWS online document
4	Application Load balancer	Routing Layer	Please refer AWS online document
5	Network Load balancer	Routing Layer	Please refer AWS online document
6	AWS VPC Peering	Routing Layer	Please refer AWS online document
7	AWS WAF	Routing Layer	Please refer AWS online document
8	AWS IAM	Common Services	Please refer AWS online document
9	AWS CloudWatch	Common Services	Please refer AWS online document
10	AWS CloudTrail	Common Services	Please refer AWS online document

Rajesh Verma

—- Confidential and not to be shared —-

11	AWS S3	Common Services	Please refer AWS online document
12	AWS Cognito	Common Services	Please refer AWS online document
13	AWS KMS	Common Services	Please refer AWS online document
14	AWS Secret Manager	Common Services	Please refer AWS online document
15	AWS SQS	Common Services	Please refer AWS online document
16	AWS SNS	Common Services	Please refer AWS online document
17	AWS SES	Common Services	Please refer AWS online document
18	AWS EC	Common Services	Please refer AWS online document
19	AWS EC2	Hosting Layer	Please refer AWS online document
20	AWS EKS	Hosting Layer	Please refer AWS online document
21	AWS RDS	Hosting Layer	Please refer AWS online document

7.5. Applicable Architecture Building Block (ABB) on GCP Cloud Platform

Seq	GCP ABB Name	AWS Service	GCP Service	ABB Layer	Description
1	Google Cloud DNS	Amazon Route 53	Google Cloud DNS	Routing Layer	Please refer GCP online document
2	Google Cloud CDN	Amazon CloudFront	Google Cloud CDN	Routing Layer	Please refer GCP online document
3	Google Cloud Endpoints	Amazon API Gateway	Google Cloud Endpoints	Routing Layer	Please refer GCP online document
4	Google Cloud Load Balancing	Application Load Balancer	Google Cloud Load Balancing	Routing Layer	Please refer GCP online document

—- Confidential and not to be shared —-

5	Google Cloud Load Balancing	Network Load Balancer	Google Cloud Load Balancing	Routing Layer	Please refer GCP online document
6	VPC Network Peering	VPC Peering	VPC Network Peering	Routing Layer	Please refer GCP online document
7	Google Cloud Armor	AWS WAF	Google Cloud Armor	Routing Layer	Please refer GCP online document
8	Google Identity and Access Management	AWS Identity and Access Management	Google Identity and Access Management	Common Services	Please refer GCP online document
9	Google Stackdriver Monitoring	Amazon CloudWatch	Google Stackdriver Monitoring	Common Services	Please refer GCP online document
10	Google Cloud Audit Logs	AWS CloudTrail	Google Cloud Audit Logs	Common Services	Please refer GCP online document
11	Google Cloud Storage	Amazon S3	Google Cloud Storage	Common Services	Please refer GCP online document
12	Firebase Authenticatio n	Amazon Cognito	Firebase Authentication	Common Services	Please refer GCP online document
13	Google Cloud Key Management	AWS Key Management Service	Google Cloud Key Management	Common Services	Please refer GCP online document
14	Google Secret Manager	AWS Secrets Manager	Google Secret Manager	Common Services	Please refer GCP online document
15	Google Cloud Pub/Sub	Amazon Simple Queue Service	Google Cloud Pub/Sub	Common Services	Please refer GCP online document
16	Google Cloud Pub/Sub	Amazon Simple	Google Cloud Pub/Sub	Common Services	Please refer GCP online document

Rajesh Verma

—- Confidential and not to be shared —-

		Notification Service			
17	None (Third-party service required)	Amazon Simple Email Service	None (Third-party service required)	Common Services	Please refer GCP online document
18	Google Kubernetes Engine (GKE)	AWS Elastic Container Service	Google Kubernetes Engine (GKE)	Common Services	Please refer GCP online document
19	Google Compute Engine	Amazon EC2	Google Compute Engine	Hosting Layer	Please refer GCP online document
20	Google Kubernetes Engine (GKE)	Amazon Elastic Kubernetes Service	Google Kubernetes Engine (GKE)	Hosting Layer	Please refer GCP online document
21	Google Cloud SQL	Amazon Relational Database Service	Google Cloud SQL	Hosting Layer	Please refer GCP online document

7.6. Applicable Architecture Building Block (ABB) on Azure Cloud Platform

Seq	ABB Name	AWS Service	Azure Service	ABB Layer	Description
1	Azure DNS	Amazon Route 53	Azure DNS	Routing Layer	Please refer Azure online document
2	Azure CDN	Amazon CloudFront	Azure CDN	Routing Layer	Please refer Azure online document

Rajesh Verma

—- Confidential and not to be shared —-

3	Azure API Management	Amazon API Gateway	Azure API Management	Routing Layer	Please refer Azure online document
4	Azure Application Gateway	Application Load Balancer	Azure Application Gateway	Routing Layer	Please refer Azure online document
5	Azure Load Balancer	Network Load Balancer	Azure Load Balancer	Routing Layer	Please refer Azure online document
6	VNet Peering	VPC Peering	VNet Peering	Routing Layer	Please refer Azure online document
7	Azure Web Application Firewall (WAF)	AWS WAF	Azure Web Application Firewall (WAF)	Routing Layer	Please refer Azure online document
8	Azure Active Directory	AWS Identity and Access Management	Azure Active Directory	Common Services	Please refer Azure online document
9	Azure Monitor	Amazon CloudWatch	Azure Monitor	Common Services	Please refer Azure online document
10	Azure Activity Logs	AWS CloudTrail	Azure Activity Logs	Common Services	Please refer Azure online document
11	Azure Blob Storage	Amazon S3	Azure Blob Storage	Common Services	Please refer Azure online document
12	Azure Active Directory B2C	Amazon Cognito	Azure Active Directory B2C	Common Services	Please refer Azure online document
13	Azure Key Vault	AWS Key Management Service	Azure Key Vault	Common Services	Please refer Azure online document
14	Azure Key Vault (Secrets)	AWS Secrets Manager	Azure Key Vault (Secrets)	Common Services	Please refer Azure online document
15	Azure Queue Storage	Amazon Simple Queue Service	Azure Queue Storage	Common Services	Please refer Azure online document

Rajesh Verma

—- Confidential and not to be shared —-

16	Azure Service Bus	Amazon Simple Notification Service	Azure Service Bus	Common Services	Please refer Azure online document
17	Azure Communication Services (Email)	Amazon Simple Email Service	Azure Communication Services (Email)	Common Services	Please refer Azure online document
18	Azure Kubernetes Service (AKS)	AWS Elastic Container Service	Azure Kubernetes Service (AKS)	Common Services	Please refer Azure online document
19	Azure Virtual Machines	Amazon EC2	Azure Virtual Machines	Hosting Layer	Please refer Azure online document
20	Azure Kubernetes Service (AKS)	Amazon Elastic Kubernetes Service	Azure Kubernetes Service (AKS)	Hosting Layer	Please refer Azure online document
21	Azure SQL Database	Amazon Relational Database Service	Azure SQL Database	Hosting Layer	Please refer Azure online document

7.7. Applicable Architecture Building Block (ABB) @ On-premises

Equivalent [**Open-Source Tool**] name for on-premises. These tools are just sample names, one can decide tools selection as per their preference and write equivalent ABB to provision them.

Seq	AWS ABB Name	AWS Service	Open-Source Tool	Description
1	AWS Route 53	Amazon Route 53	**BIND**	BIND is a widely used open-source DNS server.

Rajesh Verma

—- Confidential and not to be shared —-

2	AWS CloudFront	Amazon CloudFront	**Apache Traffic Server**	A high-performance caching proxy server capable of handling large-scale deployments.
3	AWS API-GW	Amazon API Gateway	**Kong**	Kong is a scalable, open-source API Gateway and Microservices Management Layer.
4	Application Load Balancer	Application Load Balancer	**HAProxy**	HAProxy is a reliable, high-performance TCP/HTTP load balancer.
5	Network Load Balancer	Network Load Balancer	**Nginx**	Nginx is a popular open-source web server and load balancer.
6	AWS VPC Peering	VPC Peering	**OpenVPN**	OpenVPN is an open-source VPN solution for secure network peering.
7	AWS WAF	AWS WAF	**ModSecurity**	ModSecurity is an open-source web application firewall.
8	AWS IAM	AWS Identity and Access Management	**Keycloak**	Keycloak is an open-source identity and access management solution.
9	AWS CloudWatch	Amazon CloudWatch	**Prometheus**	Prometheus is an open-source monitoring system and time series database.
10	AWS CloudTrail	AWS CloudTrail	**Auditd**	Auditd is the Linux Audit Daemon for tracking system calls and events.
11	AWS S3	Amazon S3	**MinIO**	MinIO is a high-performance, S3-compatible object storage system.
12	AWS Cognito	Amazon Cognito	**Gluu**	Gluu Server is an open-source platform for identity management.
13	AWS KMS	AWS Key Management Service	**HashiCorp Vault**	Vault is an open-source tool for secrets management, encryption, and access management.

Rajesh Verma

—- Confidential and not to be shared —-

14	AWS Secret Manager	AWS Secrets Manager	**HashiCorp Vault**	Vault is an open-source tool for secrets management, encryption, and access management.
15	AWS SQS	Amazon Simple Queue Service	**RabbitMQ**	RabbitMQ is an open-source message broker that implements the Advanced Message Queuing Protocol (AMQP).
16	AWS SNS	Amazon Simple Notification Service	**Apache Kafka**	Kafka is an open-source stream-processing software platform.
17	AWS SES	Amazon Simple Email Service	**Postal**	Postal is an open-source mail delivery platform.
18	AWS EC	AWS Elastic Container Service	**Docker Swarm**	Docker Swarm is a native clustering and scheduling tool for Docker containers.
19	AWS EC2	Amazon EC2	**OpenStack**	OpenStack is an open-source cloud computing platform for public and private clouds.
20	AWS EKS	Amazon Elastic Kubernetes Service	**Kubernetes**	Kubernetes is an open-source container orchestration system for automating application deployment, scaling, and management.
21	AWS RDS	Amazon Relational Database Service	**PostgreSQL**	PostgreSQL is an open-source relational database management system.

Rajesh Verma

—- Confidential and not to be shared —-

8. CMI : Example E2E Architecture & ABB templates

8.1. List of [On-premises-ABBs]

8.1.1. ABB : Database <MySql> @ On-premises
8.1.2. ABB : Database <Postgres> @ On-premises
8.1.3. ABB : Cache <Redis> @ On-premises
8.1.4. ABB : Cache @ On-premises
8.1.5. ABB : Observability @ On-premises
8.1.6. ABB : Load Balancer @ On-premises

8.2. List of [AWS-ABBs]

8.2.1. ABB : Database <MySql> @ AWS
8.2.2. ABB : Database <Postgres> @ AWS
8.2.3. ABB : Cache <Redis> @ AWS
8.2.4. ABB : Cache @ AWS
8.2.5. ABB : Observability @ AWS
8.2.6. ABB : Load Balancer @ AWS

8.3. List of [Azure-ABBs]

8.3.1. ABB : Database <MySql> @ Azure
8.3.2. ABB : Database <Postgres> @ Azure
8.3.3. ABB : Cache <Redis> @ Azure
8.3.4. ABB : Cache @ Azure
8.3.5. ABB : Observability @ Azure
8.3.6. ABB : Load Balancer @ Azure

Rajesh Verma

—- Confidential and not to be shared —-

8.4. List of [Azure-ABBs]

8.4.1. ABB : Database <MySql> @ Azure

8.4.2. ABB : Database <Postgres> @ Azure

8.4.3. ABB : Cache <Redis> @ Azure

8.4.4. ABB : Cache @ Azure

8.4.5. ABB : Observability @ Azure

8.4.6. ABB : Load Balancer @ Azure

Rajesh Verma

—- Confidential and not to be shared —-

9. CMI : ABB Cartridge preparation approach

1) Setup exercise for each ABB on planned deployment-platform will be performed
2) During the setup process of each ABB, NFR principles will be assessed e.g. HA/DR, Scalability, Data-loss, Storage, Connectivity, Security, Authentication, Observability etc.
3) For each NFR or setup step, there will be some input to on deployment platform
4) All those steps will be configured as “**DECOMPOSITION-TASK**” for each template
5) ABB Cartridge is analogous to Service Provisioning Cartridge in Telecom world, which are executed on Telecom Network Elements.
6) “**DECOMPOSITION-TASK**” will consists of COMMAND & Arguments, which shall be executed in “Provisioning Systems”

Rajesh Verma

—- Confidential and not to be shared —-

10. Use-cases : CMI

Seq	Use case name	Function name	Actors	Description
1	Sign-up by email	User Management	Buyer's IT team, CMI's Support team	Customers who want to set up the infrastructure, shall be registered on the CMI platform. (User and Customer both will be created on sign-up)
2	Sign-in by email	User Management	Buyer's IT team, CMI's Support team	*** Self explanatory ***
3	User Logout	User Management	Buyer's IT team, CMI's Support team	*** Self explanatory ***
4	Change Password	User Management	Buyer's IT team, CMI's Support team	*** Self explanatory ***
5	Reset Password	User Management	Buyer's IT team, CMI's Support team	*** Self explanatory ***
6	Resend OTP	User Management	Buyer's IT team, CMI's Support team	*** Self explanatory ***
7	Sign-up by mobile-num	User Management	Buyer's IT team, CMI's Support team	*** Self explanatory ***
8	Sign-up by customer-id	User Management	Buyer's IT team, CMI's Support team	*** Self explanatory ***
9	Sign-up by other identity	User Management	Buyer's IT team, CMI's Support team	*** Self explanatory ***
10	Sign-in by mobile-num	User Management	Buyer's IT team, CMI's Support team	*** Self explanatory ***
11	Sign-in by customer-id	User Management	Buyer's IT team, CMI's Support team	*** Self explanatory ***

Rajesh Verma

—- Confidential and not to be shared —-

12	Sign-in by other identity	User Management	Buyer's IT team, CMI's Support team	*** Self explanatory ***
13	Change sign-in mobile-num	User Management	Buyer's IT team, CMI's Support team	*** Self explanatory ***
14	Change sign-in email	User Management	Buyer's IT team, CMI's Support team	*** Self explanatory ***
15	Change sign-in other identity	User Management	Buyer's IT team, CMI's Support team	*** Self explanatory ***
16	Create Customer	Customer Management	Buyer's IT team	*** Self explanatory ***
17	Edit Customer (Demographic Info)	Customer Management	Buyer's IT team	*** Self explanatory ***
18	Edit Customer (Address & Contact Info)	Customer Management	Buyer's IT team	*** Self explanatory ***
19	Edit Customer (Payment Info)	Customer Management	Buyer's IT team	*** Self explanatory ***
20	Search Customer	Customer Management	Buyer's IT team	*** Self explanatory ***
21	List Customers	Customer Management	Buyer's IT team	*** Self explanatory ***
22	Disable Customer	Customer Management	Buyer's IT team	*** Self explanatory ***
23	Enable Customer	Customer Management	Buyer's IT team	*** Self explanatory ***
24	Remove customer	Customer Management	Buyer's IT team	*** Self explanatory ***
25	Manage	Architecture	Buyer's IT team	

Rajesh Verma

—- Confidential and not to be shared —-

	Environments	Provisioning System		
26	Manage Architecture Building Block (ABB)	Architecture Provisioning System	Buyer's IT team	
26	Manage Deployment Platforms	Architecture Provisioning System	Buyer's IT team	Cloud platform (AWS, GCP, Azure, etc.) On-premises
27	Manage Architecture Templates	Architecture Provisioning System	Buyer's IT team	1. Base architecture (BAS-NON-PROD) with mandatory ABBs 2. Advance architecture (ADV-NON-PROD) with mandatory and all good-to-have ABBs 3. Base architecture (BAS-PROD) with mandatory ABBs 4. Advance architecture (ADV-PROD) with mandatory and all good-to-have ABBs
28	Provision Infrastructure	Architecture Provisioning System	Buyer's IT team	**User Interface** 1) Select Environment 2) Select Deployment Platform 3) Select Architecture Template 4) Submit for Provisioning
29	Provision ABB	Architecture Provisioning System	Buyer's IT team	**User Interface** 1) Select Environment 2) Select Deployment Platform 3) Select Architecture Template 4) Select ABB 5) Submit for Provisioning
30	Manage Product Profile (Specification & Attributes)	Product Catalog	CMI's Backend Team	**a) Architecture template as Bundled product** **b)**
31	Define product prices	Product Catalog	CMI's Backend Team	
32		Product Catalog	CMI's Backend Team	

—- Confidential and not to be shared —-

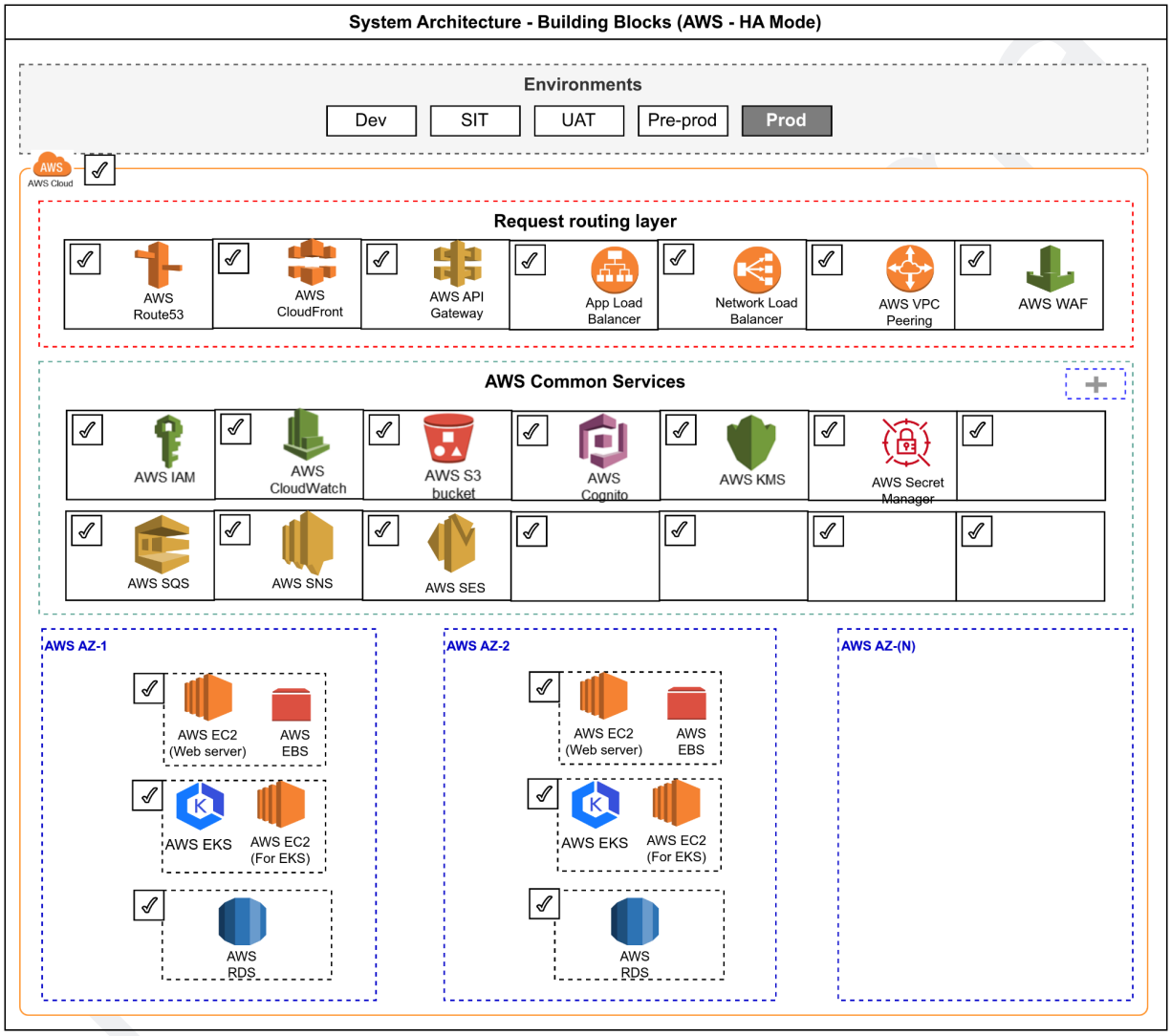
33		Sales Order Management		
34		Service (Technical) Order management		
35		Provisioning Order Management		
36		Subscription management		
37		Notification manager		
38		Service Request (Complain) Management		
39		Self-care Web-app		
40		Self-care Mobile-app		
41		Billing & Accounting		
42		Payment & Collection		
43		Architecture Diagram Generator		
44		Monitoring & Dashboard (Observability)		
45		Audit Logs		
46		Suggestion & Recommendation (AI Engine)		

Rajesh Verma

—- Confidential and not to be shared —-

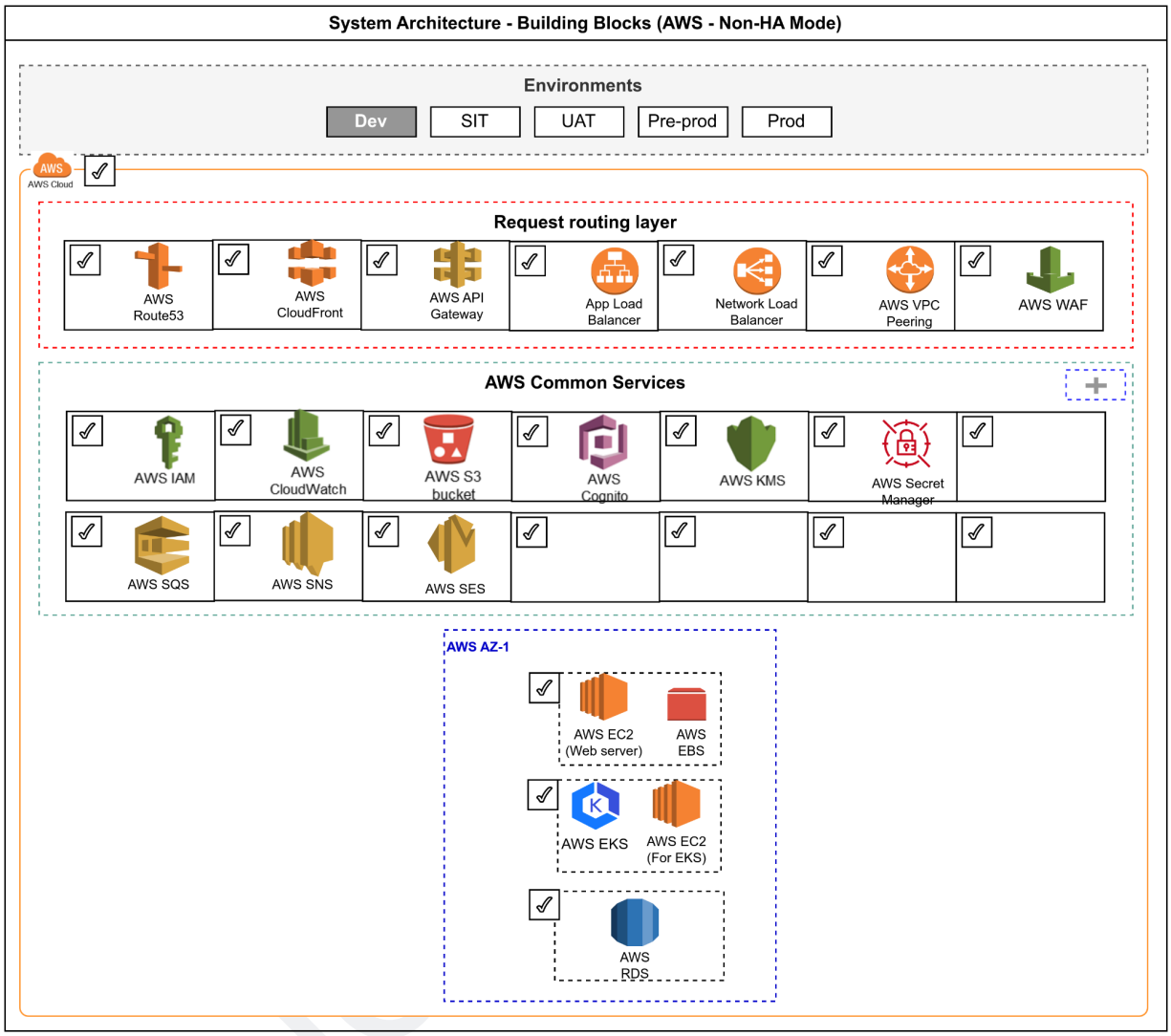
11. Outcome System Architecture (for customer)

11.1. System Architecture (AWS Production)



—- Confidential and not to be shared —-

11.2. System Architecture (AWS Non-Production)



11.3. System Architecture (GCP)

<<Architecture diagram to be added>>

11.4. System Architecture (Azure)

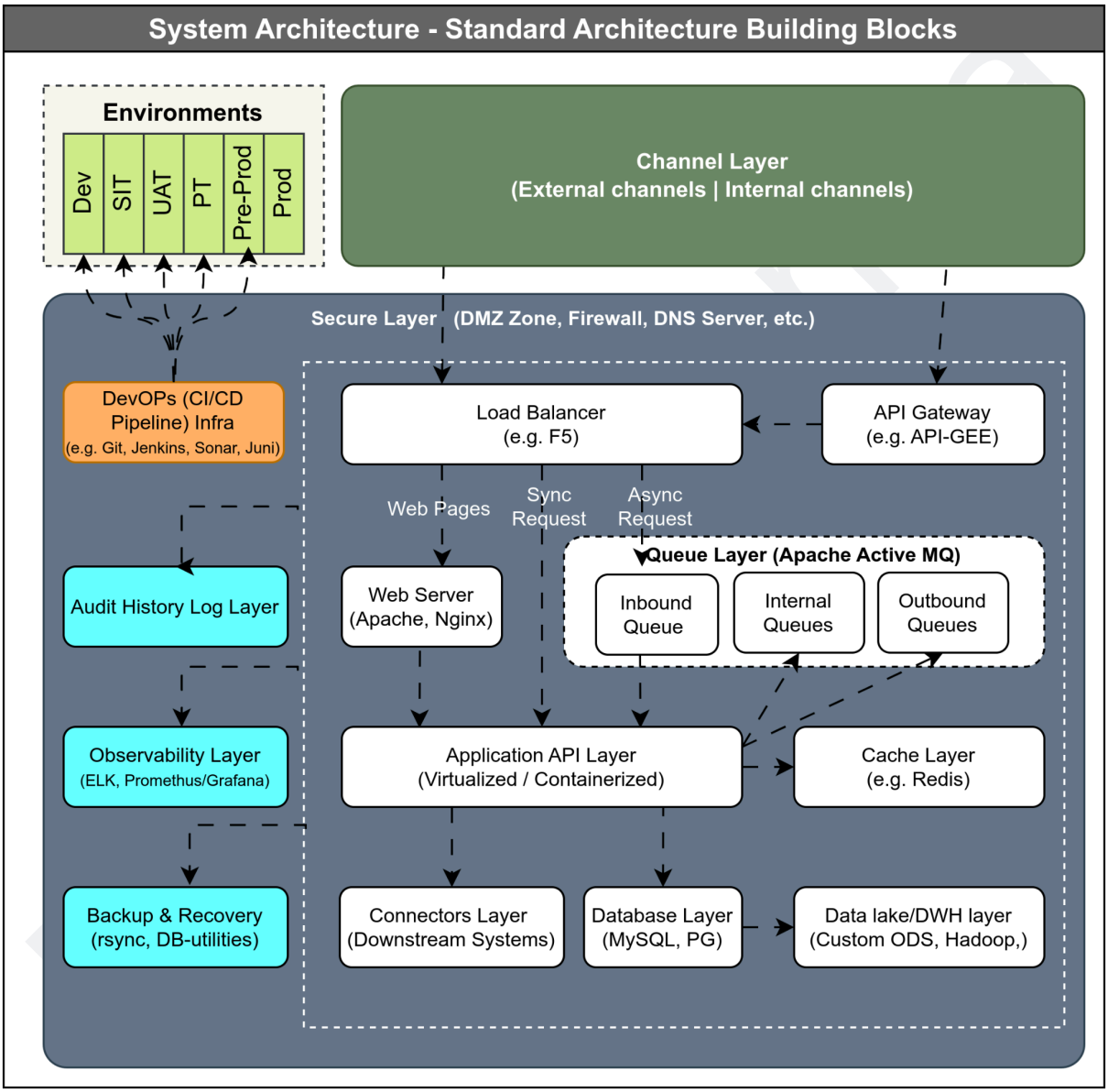
<<Architecture diagram to be added>>

—- Confidential and not to be shared —-

11.5. System Architecture (Others e.g. Oracle, RedHat, etc.)

<<Architecture diagram to be added>>

11.6. System Architecture Generic



—- Confidential and not to be shared —-

12. Outcome : Architecture TCO (Roadmap)

<<Conceptual Approach>>

1) BoM will be prepared automatically
2) Customer will capture unit-rate for each item type
3) Based on Qty x Unit-rate total TCO will be calculated
4) TCO can also have a one-time, recurring & usage charge.
5) TCO itself is a big topic, which may not be part of MVP.

13. Solution Approach : CMI

13.1. Manage Environments : Solution approach

Seq	Environment Name	Description
1	DIT	Development Integration Test
2	SIT	System Integration Test
3	UAT	User Acceptance Test
4	Pre-Prod	Pre production
5	PROD	Production

13.2. Manage Architecture Building Block (ABB) : Solution approach

Seq	ABB Name	ABB Layer	Description
1	AWS Route 53	Routing Layer	ABB Cartridge code for AWS service
2	AWS CloudFront	Routing Layer	ABB Cartridge code for AWS service
3	AWS API-GW	Routing Layer	ABB Cartridge code for AWS service
4	Application Load balancer	Routing Layer	ABB Cartridge code for AWS service
5	Network Load balancer	Routing Layer	ABB Cartridge code for AWS service

Rajesh Verma

—- Confidential and not to be shared —-

6	AWS VPC Peering	Routing Layer	ABB Cartridge code for AWS service
7	AWS WAF	Routing Layer	ABB Cartridge code for AWS service
8	AWS IAM	Common Services	ABB Cartridge code for AWS service
9	AWS CloudWatch	Common Services	ABB Cartridge code for AWS service
10	AWS CloudTrail	Common Services	ABB Cartridge code for AWS service
11	AWS S3	Common Services	ABB Cartridge code for AWS service
12	AWS Cognito	Common Services	ABB Cartridge code for AWS service
13	AWS KMS	Common Services	ABB Cartridge code for AWS service
14	AWS Secret Manager	Common Services	ABB Cartridge code for AWS service
15	AWS SQS	Common Services	ABB Cartridge code for AWS service
16	AWS SNS	Common Services	ABB Cartridge code for AWS service
17	AWS SES	Common Services	ABB Cartridge code for AWS service
18	AWS EC	Common Services	ABB Cartridge code for AWS service
19	AWS EC2	Hosting Layer	ABB Cartridge code for AWS service
20	AWS EKS	Hosting Layer	ABB Cartridge code for AWS service
21	AWS RDS	Hosting Layer	ABB Cartridge code for AWS service

13.3. Manage Deployment Platforms : Solution approach

Seq	Deployment Platform	Description
1	AWS Cloud	CMI will use AWS Cloud to deploy infrastructure
2	GCP Cloud	CMI will use GCP Cloud to deploy infrastructure
3	Azure Cloud	CMI will use Azure Cloud to deploy infrastructure
4	On-premises	CMI will use On-premises setup to deploy infrastructure

Rajesh Verma

—- Confidential and not to be shared —-

5		
6		

13.4. Manage Architecture Templates : Solution approach

Seq	Environment Name	Description
1	BAS-NON-PROD	Base architecture (BAS-NON-PROD) with mandatory ABBs
2	ADV-NON-PROD	Advance architecture (ADV-NON-PROD) with mandatory and all good-to-have ABBs
3	BAS-PROD	Base architecture (BAS-PROD) with mandatory ABBs
4	ADV-PROD	Advance architecture (ADV-PROD) with mandatory and all good-to-have ABBs

Rajesh Verma

—- Confidential and not to be shared —-

14. Provision Infrastructure : Solution approach

14.1. Provision Infrastructure : User Interface (HA Mode)

Provision Architecture (Production)

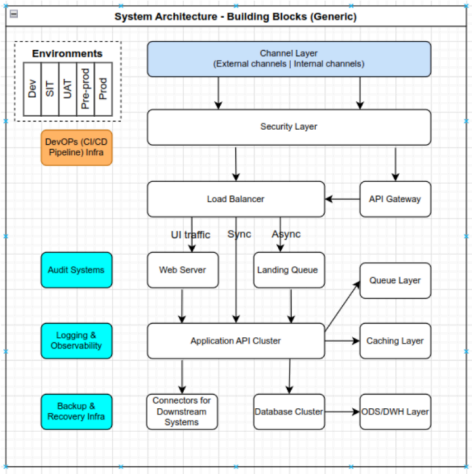
Select environment type (e.g. Prod)

Select deployment platform (e.g. AWS)

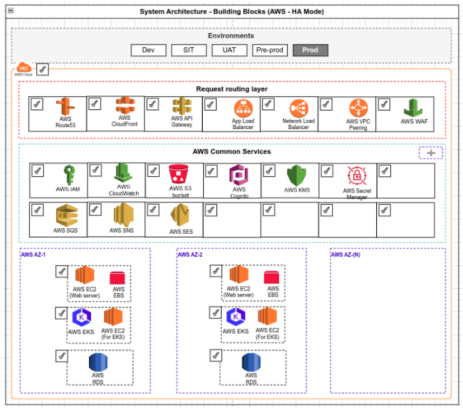
Select architecture template (e.g. BAS-PROD)

Submit to Provision

Preview - Generic

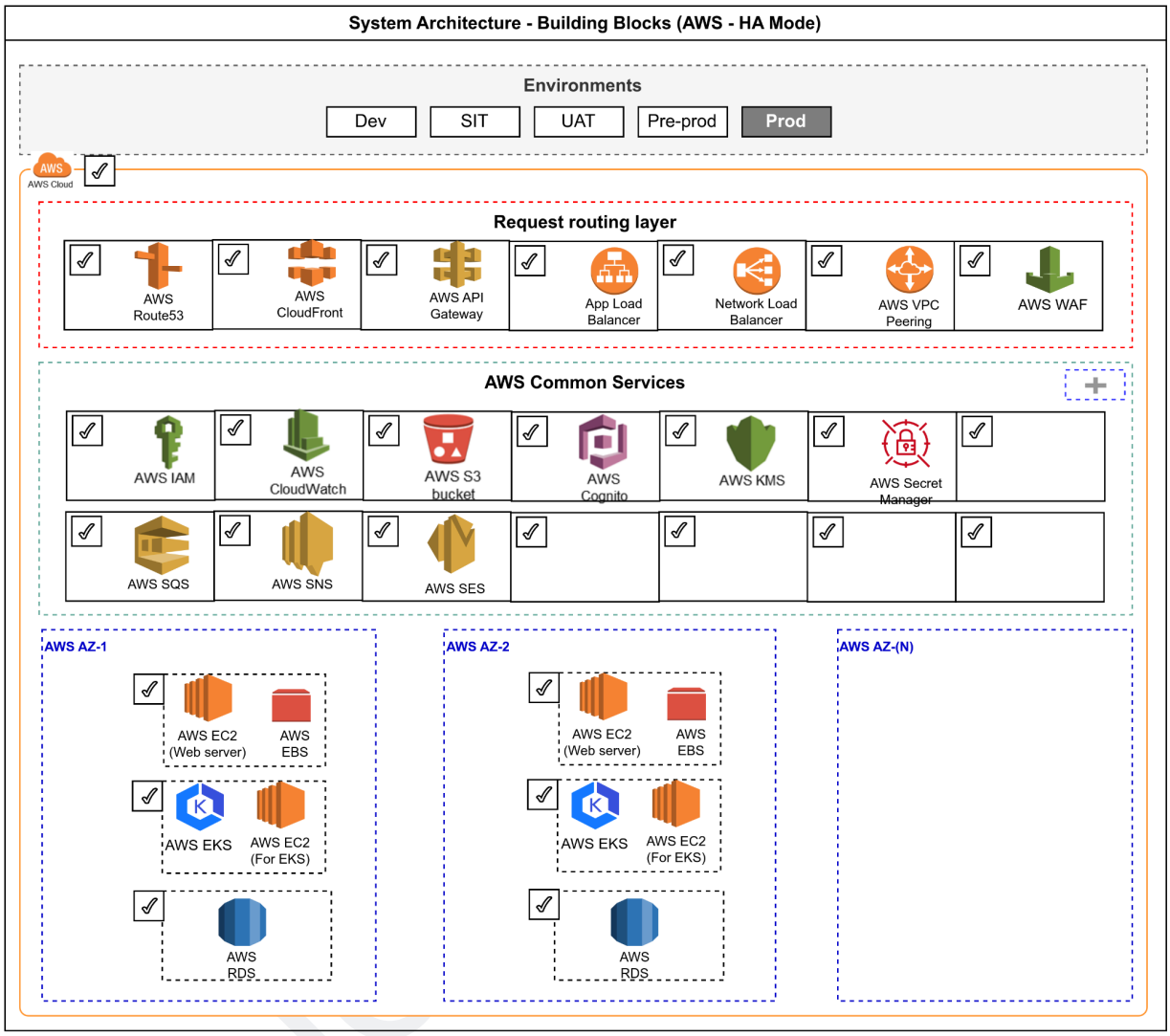


Preview - AWS



—— Confidential and not to be shared ——

14.2. Preview of provisioned infrastructure (HA Mode)



—- Confidential and not to be shared —-

14.3. Provision Infrastructure : User Interface (Non-HA Mode)

Provision Architecture (Non-Prod)

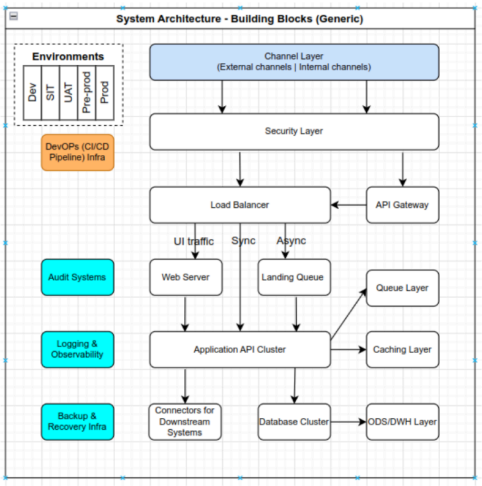
Select environment type (e.g. Dev)

Select deployment platform (e.g. AWS)

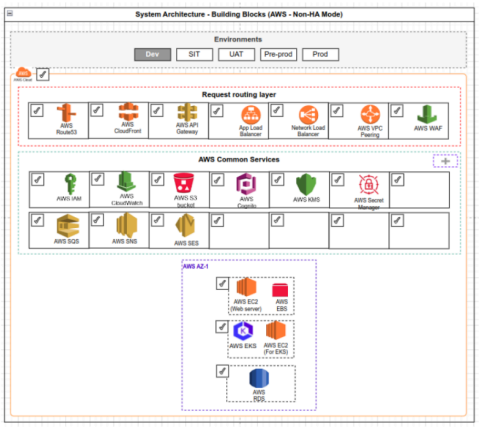
Select architecture template (e.g. BAS-NON-PROD)

Submit to Provision

Preview - Generic

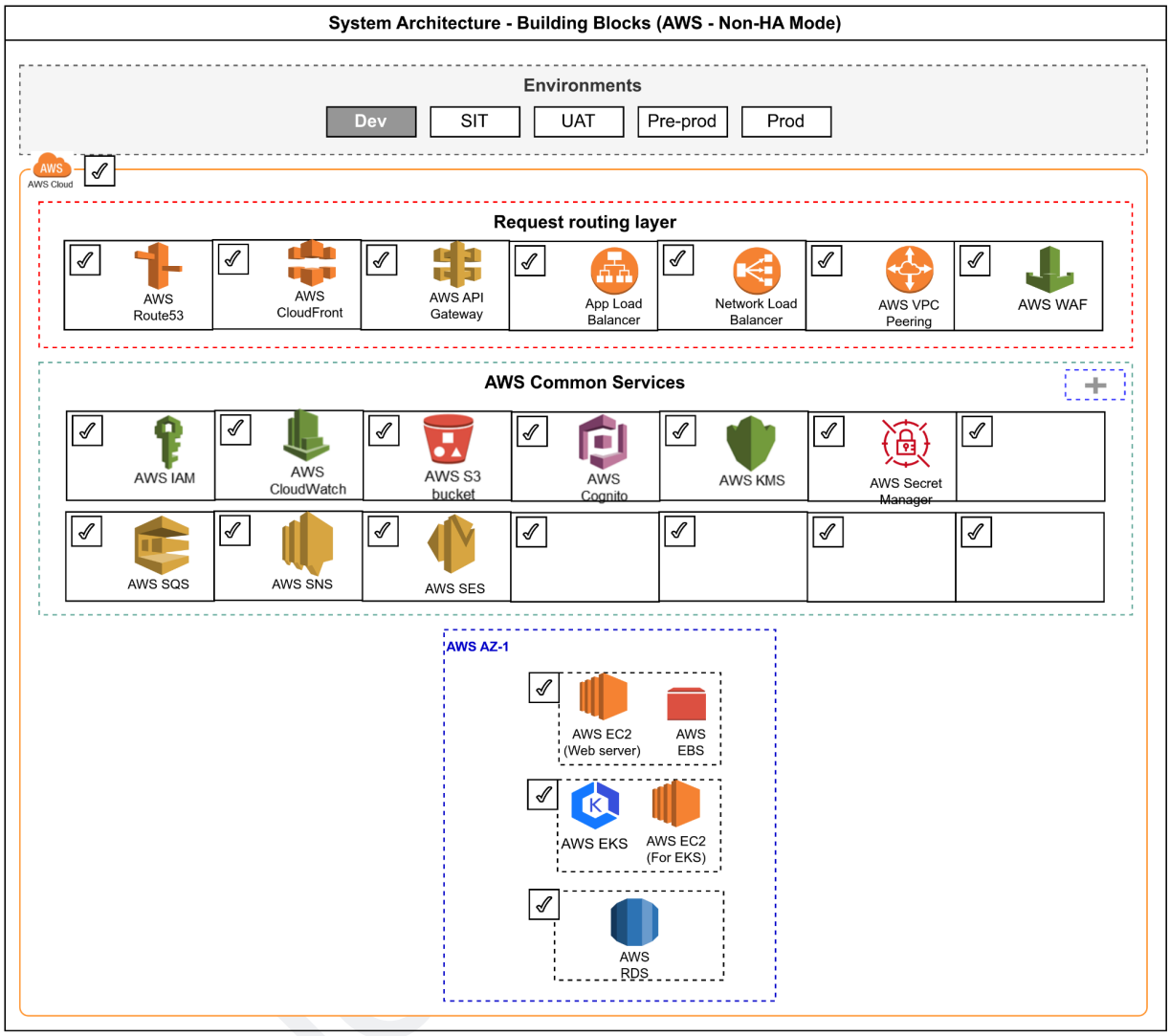


Preview - AWS



—— Confidential and not to be shared ——

14.4. Preview of provisioned infrastructure (Non-HA Mode)



14.5. Provision Infrastructure : High Level Design

<<To be added>>

—- Confidential and not to be shared —-

15. De-provision ABB : Solution approach

15.1. De-provision ABB : User Interface

Deprovision Architecture Building Block (ABB)

Select environment type (e.g. Prod)

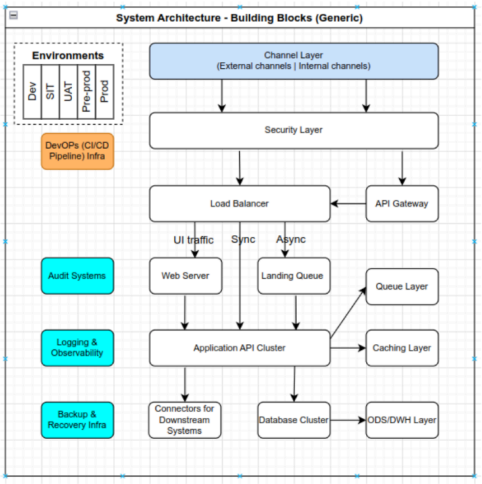
Select deployment platform (e.g. AWS)

Select architecture template (e.g. BAS-PROD)

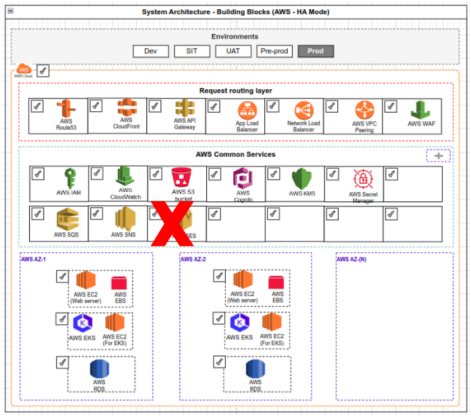
Select architecture building block (e.g. SES)

Submit to Deprovision

Preview - Generic



Preview - AWS



15.2. Deprovision Infrastructure : High Level Design

<<To be added>>

—— Confidential and not to be shared ——

16. Provision ABB : Solution approach

16.1. Provision ABB : User Interface

Provision Architecture Building Block (ABB)

Select environment type (e.g. Prod)

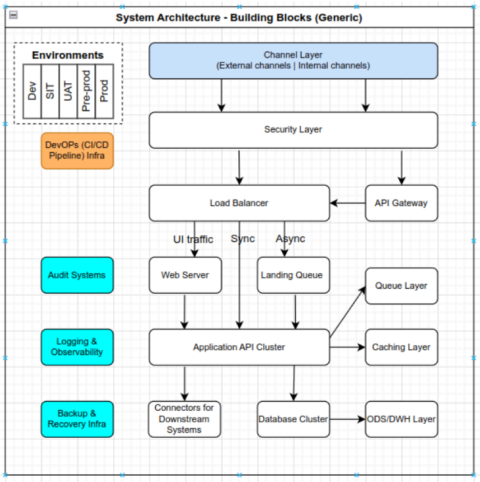
Select deployment platform (e.g. AWS)

Select architecture template (e.g. BAS-PROD)

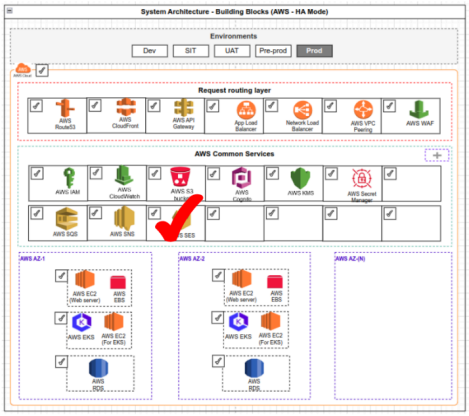
Select architecture building block (e.g. SES)

Submit to Provision

Preview - Generic



Preview - AWS



16.2. Provision Infrastructure : High Level Design

<<To be added>>

—— Confidential and not to be shared ——

17. Import Legacy Infrastructure : Solution approach

17.1. Import Legacy Infrastructure : User Interface

Import Legacy Infrastructure (One ABB One time)

Select environment type (e.g. Prod)

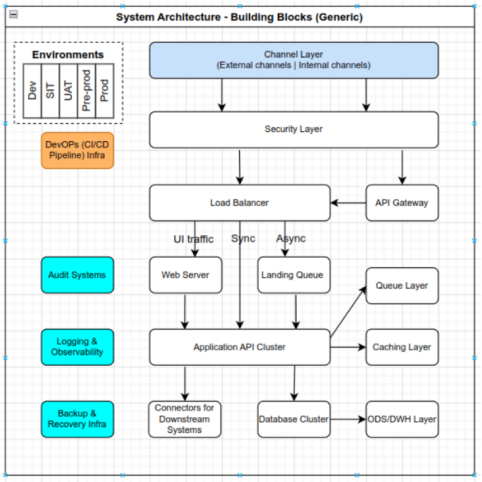
Select deployment platform (e.g. AWS)

Select architecture template (e.g. BAS-PROD)

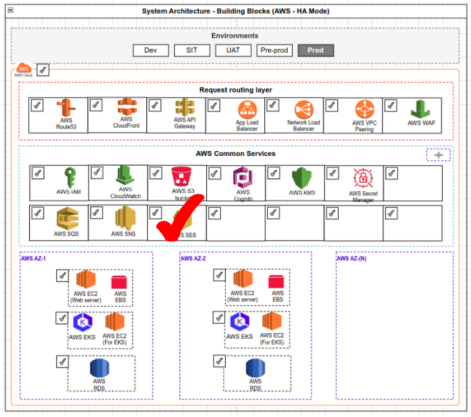
Select architecture building block (e.g. SES)

Submit to Import

Preview - Generic



Preview - AWS



17.2. Import Legacy Infrastructure : High Level Design

<<To be added>>

—— Confidential and not to be shared ——

18. Appendix

18.1. ABB Cartridges @ AWS

18.1.1. AWS Route 53

JSON	Example

	<pre>"myDNSRecord" : {
	  "Type" : "AWS::Route53::RecordSet",
	  "Properties" :
	  {
	  "HostedZoneId" : "Z3DG6IL3SJCGPX",
	  "Name" : "mysite.example.com.",
	  "Type" : "SPF",
	  "TTL" : "900",
	  "ResourceRecords" : ["\"v=spf1 ip4:192.168.0.1/16 -all\""]
	  }
	}</pre>

18.1.2. AWS CloudFront

JSON	Example

<pre>{
  "Type" : "AWS::CloudFront::Distribution",
  "Properties" : {
  "DistributionConfig" : DistributionConfig,
  "Tags" : [Tag, ...]
  }
}</pre>
	<pre>{
	  "AWSTemplateFormatVersion": "2010-09-09",
	  "Resources": {
	  "cloudfrontdistribution": {
	  "Type": "AWS::CloudFront::Distribution",
	  "Properties": {
	  "DistributionConfig": {
	  "CacheBehaviors": [
	  {
	"LambdaFunctionAssociations": [
	  {
	  "EventType": "string-value",
	"LambdaFunctionARN": "string-value"
	  }
	  ]
	  }
	  ],
	  "DefaultCacheBehavior": {
	  "LambdaFunctionAssociations": [
	  {
	  "EventType": "string-value",
	  "LambdaFunctionARN": "string-value"
	  }
	  ]
	  },
	  "IPV6Enabled": "boolean-value",
	  "Origins": [
	  {
	  "CustomOriginConfig": {
	"OriginKeepaliveTimeout": "integer-value",
	  "OriginReadTimeout": "integer-value"
	  }
	  }
	  ]</pre>

Rajesh Verma

—— Confidential and not to be shared ——

	<pre>},
	"Tags": [
	{
	"Key": "string-value",
	"Value": "string-value"
	}
	]
	}
	}
	}
	}</pre>

18.1.3. AWS API-GW

JSON	Example

<pre>{
"Type" : "AWS::ApiGateway::Method",
"Properties" : {
"ApiKeyRequired" : *Boolean*,
"AuthorizationScopes" : *[String, ...]*,
"AuthorizationType" : *String*,
"AuthorizerId" : *String*,
"HttpMethod" : *String*,
"Integration" : *Integration*,
"MethodResponses" : *[MethodResponse, ...]*,
"OperationName" : *String*,
"RequestModels" : *{Key: Value, ...}*,
"RequestParameters" : *{Key: Value, ...}*,
"RequestValidatorId" : *String*,
"ResourceId" : *String*,
"RestApiId" : *String*
}
}</pre>
	<pre>{
	"MockMethod": {
	"Type": "AWS::ApiGateway::Method",
	"Properties": {
	"RestApiId": {
	"Ref": "MyApi"
	},
	"ResourceId": {
	"Fn::GetAtt": [
	"MyApi",
	"RootResourceId"
	]
	},
	"HttpMethod": "GET",
	"AuthorizationType": "NONE",
	"Integration": {
	"Type": "MOCK"
	}
	}
	}
	}</pre>

18.1.4. Application Load balancer
18.1.5. Network Load balancer
18.1.6. AWS VPC Peering
18.1.7. AWS WAF
18.1.8. AWS IAM
18.1.9. AWS CloudWatch
18.1.10. AWS CloudTrail
18.1.11. AWS S3
18.1.12. AWS Cognito
18.1.13. AWS KMS
18.1.14. AWS Secret Manager

Rajesh Verma

—- Confidential and not to be shared —-

18.1.15. AWS Kinesis data streamAWS SQS

JSON

```
{
  "Type" : "AWS::SQS::Queue",
  "Properties" : {
      "ContentBasedDeduplication" : Boolean,
      "DeduplicationScope" : String,
      "DelaySeconds" : Integer,
      "FifoQueue" : Boolean,
      "FifoThroughputLimit" : String,
      "KmsDataKeyReusePeriodSeconds" : Integer,
      "KmsMasterKeyId" : String,
      "MaximumMessageSize" : Integer,
      "MessageRetentionPeriod" : Integer,
      "QueueName" : String,
      "ReceiveMessageWaitTimeSeconds" : Integer,
      "RedriveAllowPolicy" : Json,
      "RedrivePolicy" : Json,
      "SqsManagedSseEnabled" : Boolean,
      "Tags" : [ Tag, ... ],
      "VisibilityTimeout" : Integer
    }
}
```

Example

```
{
  "AWSTemplateFormatVersion" : "2010-09-09",

  "Description" : "This example template shows how to
create an Amazon SQS queue with CloudWatch alarms on
queue depth. This template creates an Amazon SQS queue
and one or more CloudWatch alarms. You will be billed
for the AWS resources used if you create a stack using
this template.",

  "Parameters" : {
    "AlarmEmail": {
      "Default": "jane.doe@example.com",
      "Description": "Email address to notify of
operational issues",
      "Type": "String"
    }
  },

  "Resources" : {
    "MyQueue" : {
      "Type" : "AWS::SQS::Queue",
      "Properties" : {
         "QueueName" : "SampleQueue"
      }
    },
    "AlarmTopic": {
      "Type": "AWS::SNS::Topic",
      "Properties": {
        "Subscription": [{
          "Endpoint": { "Ref": "AlarmEmail" },
          "Protocol": "email"
        }]
      }
    },
    "QueueDepthAlarm": {
      "Type": "AWS::CloudWatch::Alarm",
      "Properties": {
        "AlarmDescription": "Alarm if queue depth
increases to more than 10 messages",
        "Namespace": "AWS/SQS",
        "MetricName":
"ApproximateNumberOfMessagesVisible",
        "Dimensions": [{
          "Name": "QueueName",
          "Value" : { "Fn::GetAtt" : ["MyQueue",
"QueueName"] }
        }],
        "Statistic": "Sum",
        "Period": "300",
        "EvaluationPeriods": "1",
        "Threshold": "10",
        "ComparisonOperator": "GreaterThanThreshold",
        "AlarmActions": [{
          "Ref": "AlarmTopic"
        }],
        "InsufficientDataActions": [{
          "Ref": "AlarmTopic"
        }]
      }
    }
  },
  "Outputs" : {
    "QueueURL" : {
      "Description" : "URL of new Amazon SQS Queue",
      "Value" : { "Ref" : "MyQueue" }
    },
    "QueueARN" : {
      "Description" : "ARN of new Amazon SQS Queue",
      "Value" : { "Fn::GetAtt" : ["MyQueue", "Arn"]}
    },
    "QueueName" : {
      "Description" : "Name new Amazon SQS Queue",
      "Value" : { "Fn::GetAtt" : ["MyQueue",
```

Rajesh Verma

—— Confidential and not to be shared ——

	"QueueName"]} } } }

18.1.16. AWS SNS

JSON	Example

{
"Type" : "AWS::SNS::Topic",
"Properties" : {
"ContentBasedDeduplication" : *Boolean*,
"DataProtectionPolicy" : *Json*,
"DisplayName" : *String*,
"FifoTopic" : *Boolean*,
"KmsMasterKeyId" : *String*,
"SignatureVersion" : *String*,
"Subscription" : *[Subscription, ...]*,
"Tags" : *[Tag, ...]*,
"TopicName" : *String*,
"TracingConfig" : *String*
}
}
	"MySNSTopic" : {
	"Type" : "AWS::SNS::Topic",
	"Properties" : {
	"Subscription" : [{
	"Endpoint" : { "Fn::GetAtt" : [
	"MyQueue1", "Arn"] },
	"Protocol" : "sqs"
	}, {
	"Endpoint" : { "Fn::GetAtt" : [
	"MyQueue2", "Arn"] },
	"Protocol" : "sqs"
	}],
	"TopicName" : "SampleTopic"
	}
	}

18.1.17. AWS SES
18.1.18. AWS EC
18.1.19. AWS EC2
18.1.20. AWS EKS
18.1.21. AWS RDS

18.2. ABB Cartridges @ GCP
18.3. ABB Cartridges @ Azure
18.4. ABB Cartridges @ On-premises

Rajesh Verma

—- Confidential and not to be shared —-

19. Appendix-2

19.1. Scenarios/Rules to define ABB Cartridges

1) EKS ABB Cartridge will vary based on following scenarios
    a) EKS Cartridge for one availability zone
    b) EKS Cartridge for two availability zones
    c) EKS Cartridge for three availability zones

2) RDS ABB Cartridge will vary based on following scenario
    a) RDS Cartridge for one availability zone
    b) RDS Cartridge for two availability zones
    c) RDS Cartridge for MySQL, PostgreSQL, etc

3) Alternate option : ChatGPT can be leveraged to define script for “ABB Cartridge”

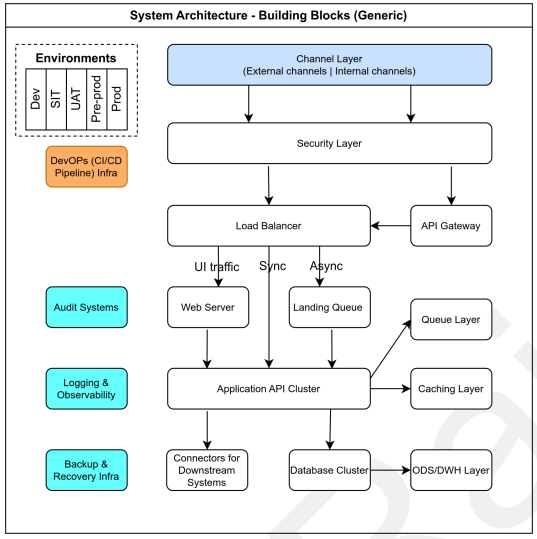
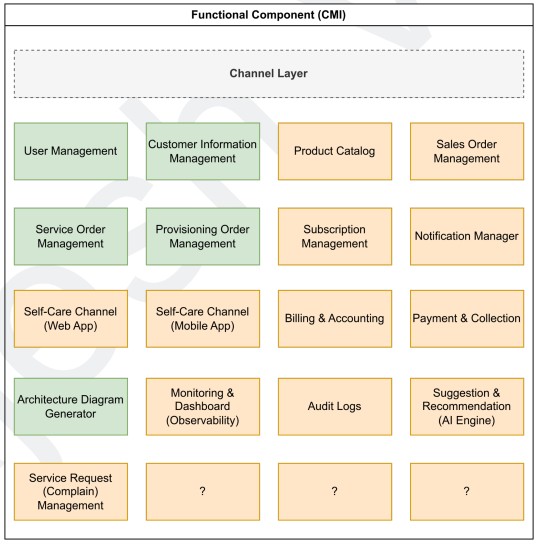
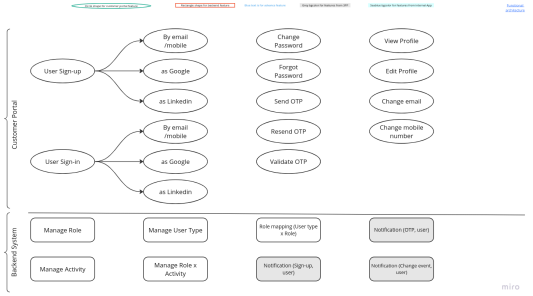
Rajesh Verma

—- Confidential and not to be shared —-

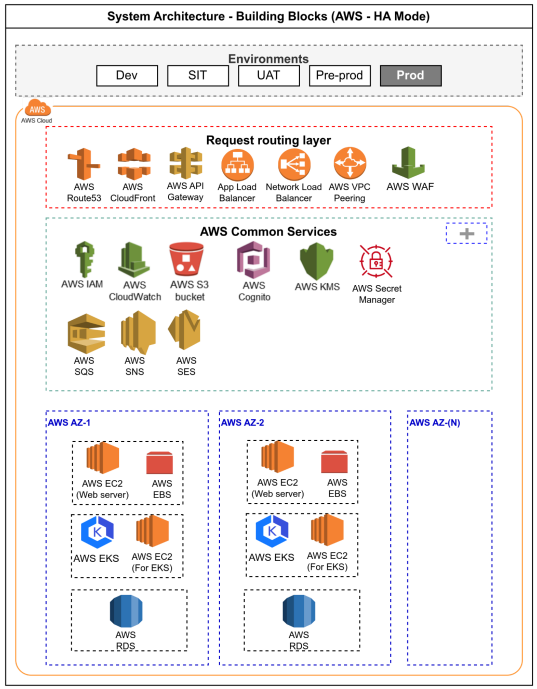
19.2. Bigger Vision : Open Capability Platform (OCP <LoCoNoCo>)

OCP (The Open Capability Platform) which has the objective to help people who wants to own the digital platform to run their business with no fear of IT knowledge.

1. Infrastructure service capability (ICS) - via CMI (Create My Infrastructure) setup
2. Business service capability (BSC) - via CMB (Create My Business) function
3. Experience development capability (xDC) - via CMUX (Create My User Experience)

CMI	CMB	CMUX
		 Please refer the zoom-in image below the table

—- Confidential and not to be shared —-



Kind of integration architecture diagram between Functional ABBs

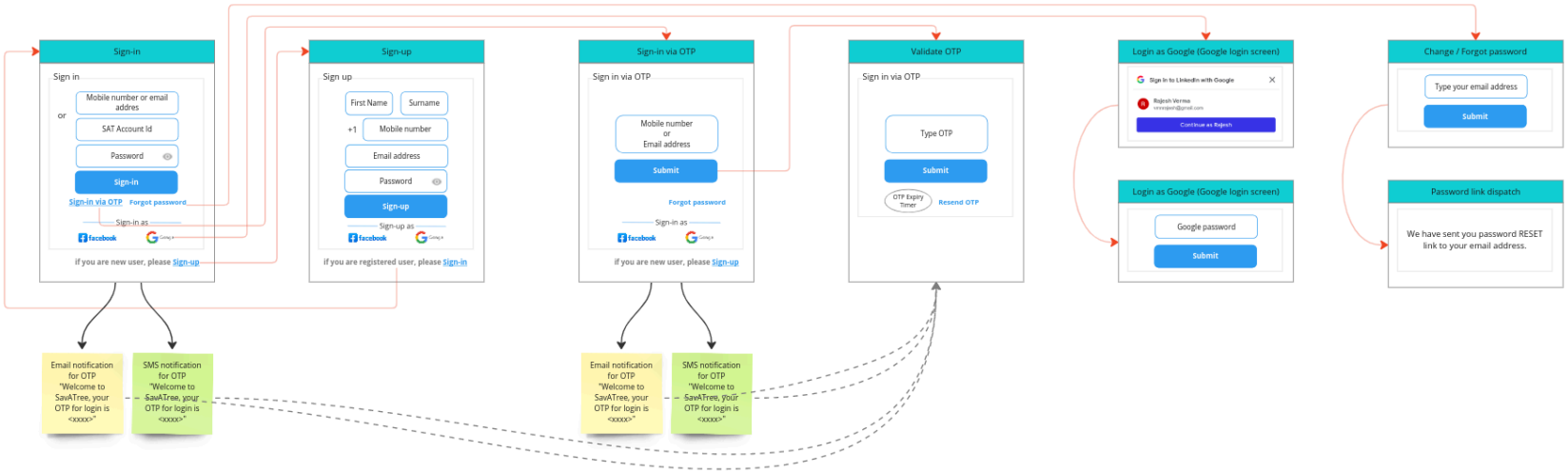


Please refer the zoom-in image below the table

Rajesh Verma

—- Confidential and not to be shared —-

19.3. Zoom-in image of CMUX

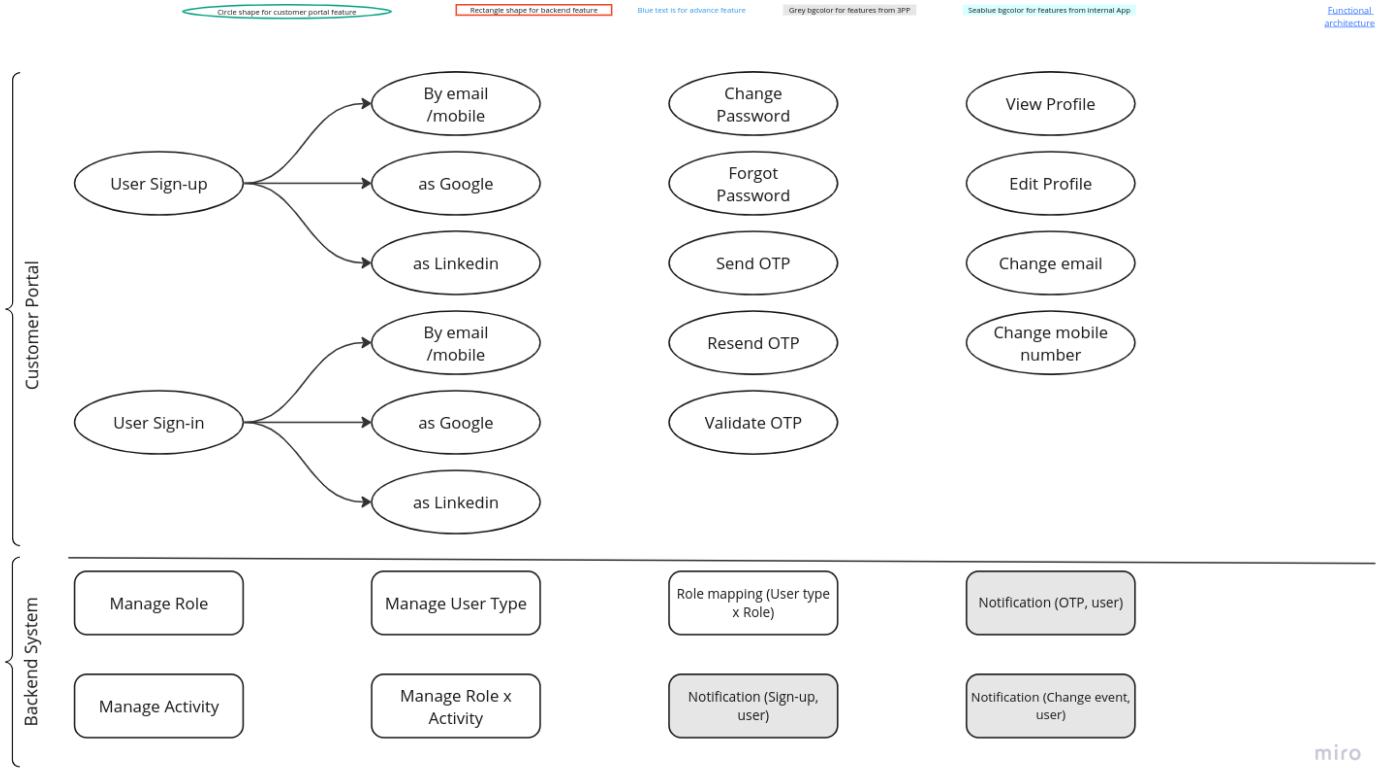


miro

Rajesh

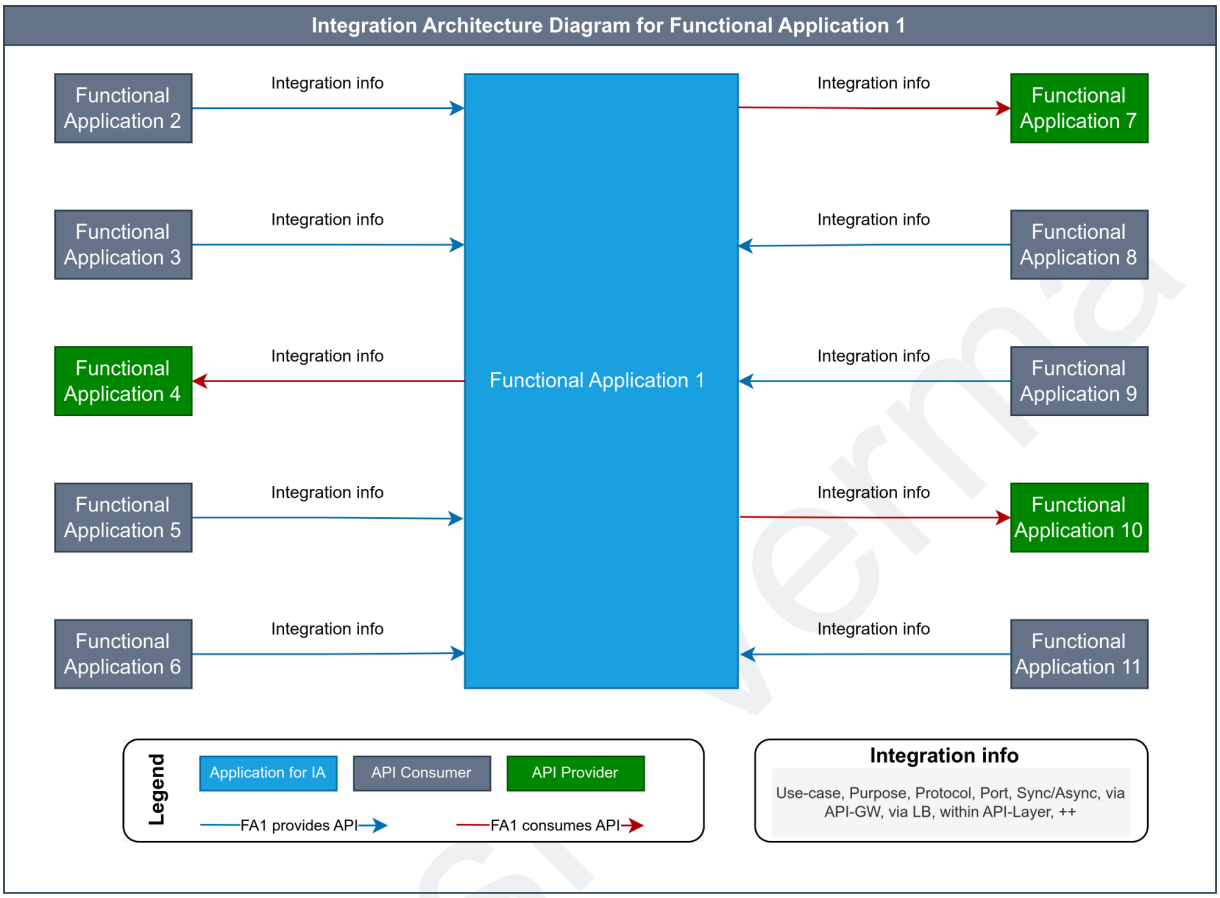
—- Confidential and not to be shared —-

19.4. Zoom-in image of CMUX



—- Confidential and not to be shared —-

19.5. Integration Architecture



In a [Large, Complex & Long] project, functional applications will communicate with each via API for different purposes. Star diagram representation will help to understand [Integration Architecture] of individual applications and the entire ecosystem collectively.

Rajesh Verma

—- Confidential and not to be shared —-

About author

Profile : Rajesh Verma - Brief profile

Source : link for this article here

Series : S1 (Architecture & design)

Episode : S1E2 ([S1-A&D] HLD : Create My Infrastructure)

Author's approach : Rajesh wants to share his learning & experience gained throughout his career from various sources. Author started the series on architectural topics and this article is one of the episodes in that attempt. Author feels that lots of information is available on various forums, but scattered here & there. Episodes in this series will be designed for most of the relevant topics in architecture-&-design, published gradually and organized in logical sequence. Principally episodes will have linkage with other episodes, so that readers can have proper connection among the topics and would be able to correlate with ongoing activities in their software life. Topics for example will be related to functional architecture, integration architecture, deployment architecture, microscopic view of mostly architecture-building-blocks (ABBs), security guidelines & approach to comply, performance KPIs & engineering, git branch & DevOps enabled automation strategy, NFR aspects (e.g. scalability, high-availability, stability, resiliency, etc.), commonly used architecture styles & design patterns, cloudification approaches, multi-tenancy approach, data migration, channel-cutover & rollout strategy, process standardization & simplification, greenfield rollout & brownfield transformation journeys, etc.

Thank you for reading the post, please stay connected.

Small note on HLD (Solution Approach)

*In order to keep the article size smaller, not sharing the entire solution-approach here. But the original document (Google Docs) will be available for all, interested & enthusiastic friends can follow the (Link of HLD document). For a long time, I was thinking of building this open-platform and [**ABB cartridge**] for each technology component, is the atomic component on the [**CMI platform**]. [**ABB cartridge**] is very much similar to [**Provisioning-cartridges-in-Telecom**] e.g. SIM-Activation-Provisioning-Cartridge, Change-SIM-Provision-Cartridge etc. for Telco operators. Technology components in any system architecture, in our architecture world is called [**Architecture Building Blocks (ABB)** (article on same here)] and [ABB-Cartridge] is the independent utility which can be download by user and execute in their infrastructure manually, or CMI platform can be allowed to executed in user's infrastructure automatically.*

Rajesh Verma

—- Confidential and not to be shared —-