**Delivering real business value of your Enterprise Architecture (EA) using “Value Streams”- Key points**

1. Have a clear vision for the Org -- Purpose & where they are headed

2. Strategy communication & implementation is a traditional approach

-- A strategy in effect results in little pieces that are picked up in different parts of the Org which is broken down into tasks, project executions, program increments and assigned to individuals at precise action oriented levels.

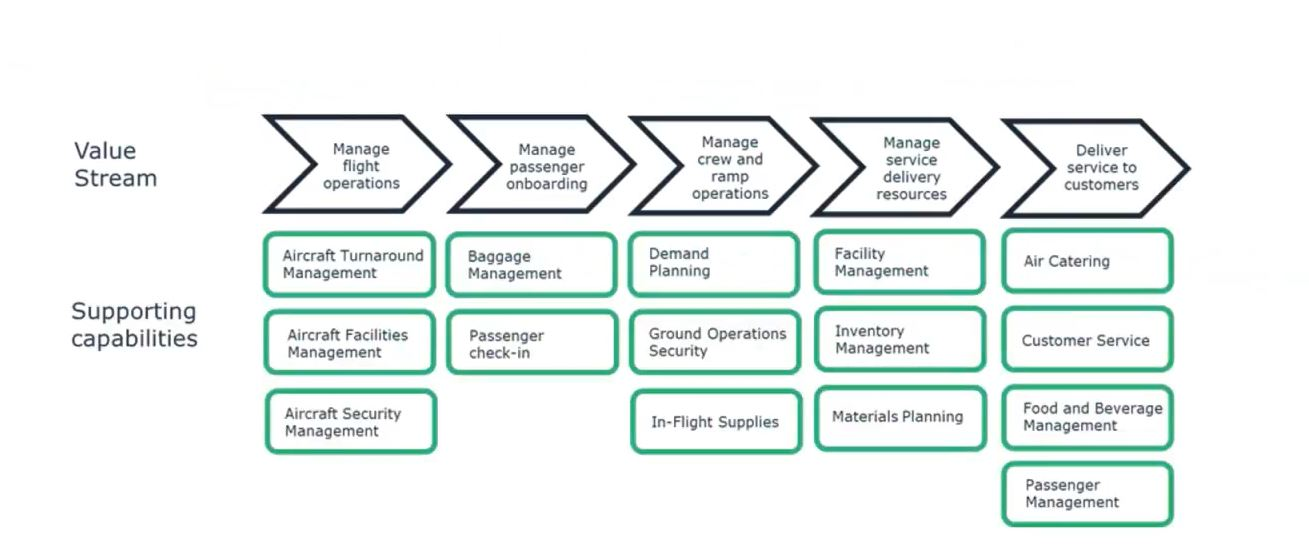
-- Ultimate alignment to the strategy may be lost

3. An EA practice aims at aligning the enterprise execution (operating model) to the company strategy. "Continuously align the strategy & operations"

4. EA & modern agile software development groups need to work together towards a common Org goal i.e Agile teams should not look upon EA as work stoppers & EA should not impose constraints on software dev teams. Instead, both should work collaboratively towards achieving the Org's goals. EA teams should help dev teams to achieve their goals.

5. EA teams should establish rules, boundaries, patterns, best practices, guardrails, success elements that will help & guide teams to achieve their goals. The greatest success will come from cooperation between the EA practice & s/w team members.

6. "Value Streams" enables to works across different architecture teams EA, Sol Architect, System architect. It helps take the strategic vision that we need to achieve, break it down into customer centric actionable approach, articulate elements around business capabilities and put it into an actual roadmap for transformation.



7. "Value Streams" business processes focuses on customer centricity and different elements that add specific value to customers. Below the value stages (business process) of the value streams will be business capabilities that are needed to fulfill the vision of the value stream

8. Now we look into the Org strategy & identify what changes in the business capabilities need to brought about to fulfill the strategy of the Org

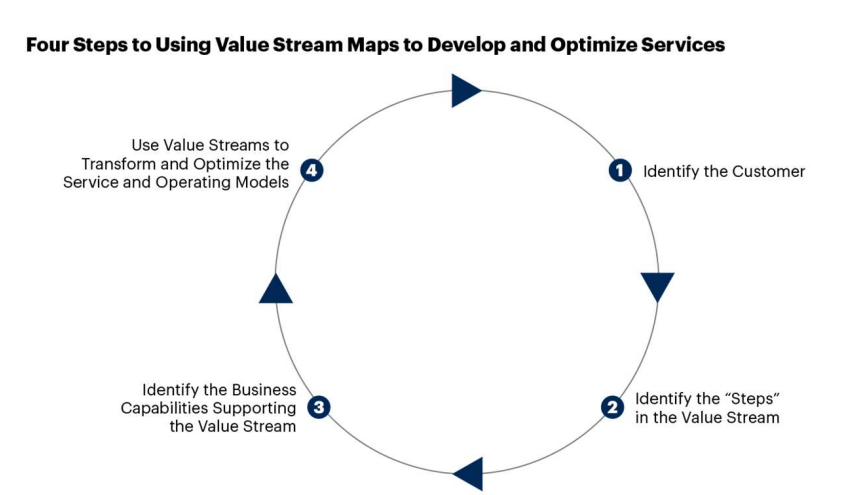
9. Value Stream Maps (VSM) uses a 4 step approach to develop & optimize services

-- Identify customer

-- Identify steps(stages) in the value stream that will provide concrete value to that customer

-- Find business capabilities underpinning these value stream stages

-- Use the value streams to transform & optimize the service & operating models



10. VSM's are essential to building, transforming & optimizing an Org's technology enabled business models.

11. VSM's provide insight & lens into how underlying applications, information & technology support the end-end value stream activities

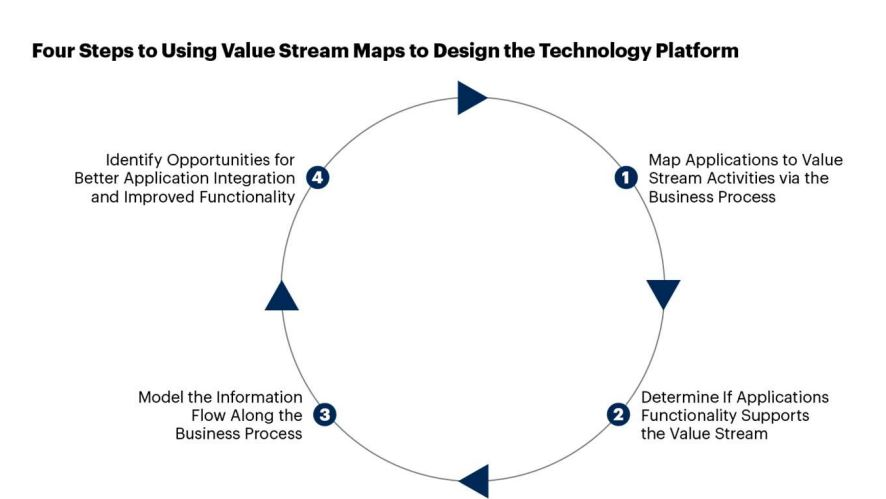
12. Once built, a value stream can be used at the enterprise level (top down) & product level (bottom up) to identify, analyze & determine the applications, information & tehnologies required in digital platforms to enable & drive the Org's future state operating model and its supporting business process.

-- Map applications to value stream activities via the business processes

-- Determine whether application functionality supports the value stream

-- Model information flow along the business process

-- Identify opportunities for better application integration and improved functionality



10. Value streams socialize knowledge across functional silos of how business capabilities interact to drive value to the customer

11. Outcomes from capabilities

-- Metrics can be attached to capabilities that can measure whether the capability is underperforming wrt Org strategy

-- To fulfill the Org strategy, what changes in People, Process & Technology need to brought about to improve the metrics & enhance the capability

-- By doing this, strategic value of software that really demonstrates customer value. This can be a differentiator between One org with another.

12. Role of EA is to enable, accelerate & facilitate using

-- Standards, principles, patterns, controls. Bring in commonality eg security of common services

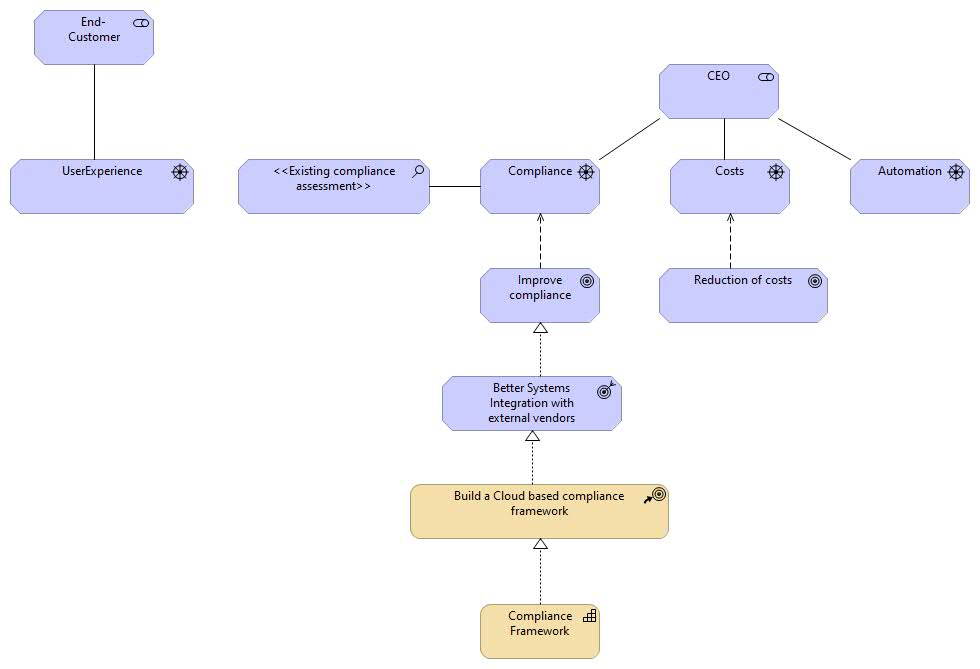
-- Provide a path to developers

13. AN EA in the modern Agile world may just use enough Architecture using EA frameworks like TOGAF. Instead of following an entire ADM lifecycle, minimal architecture required for Agile teams to start implementation is the way to proceed

**Value Streams Implementation Approach**

1. Use EA framework tools like TOGAF content framework to develop the Architectural Building Blocks (ABB's). The major ABB's are described below

-- A motivation model built using the content framework's meta-model elements. These Architectural Views will represent the Architecture Vision based on the Org's strategy. Measurable business outcomes for Org goals & its corresponding course of action resulting in creation of new business capabilities / updation of existing business capabilities can be depicted.

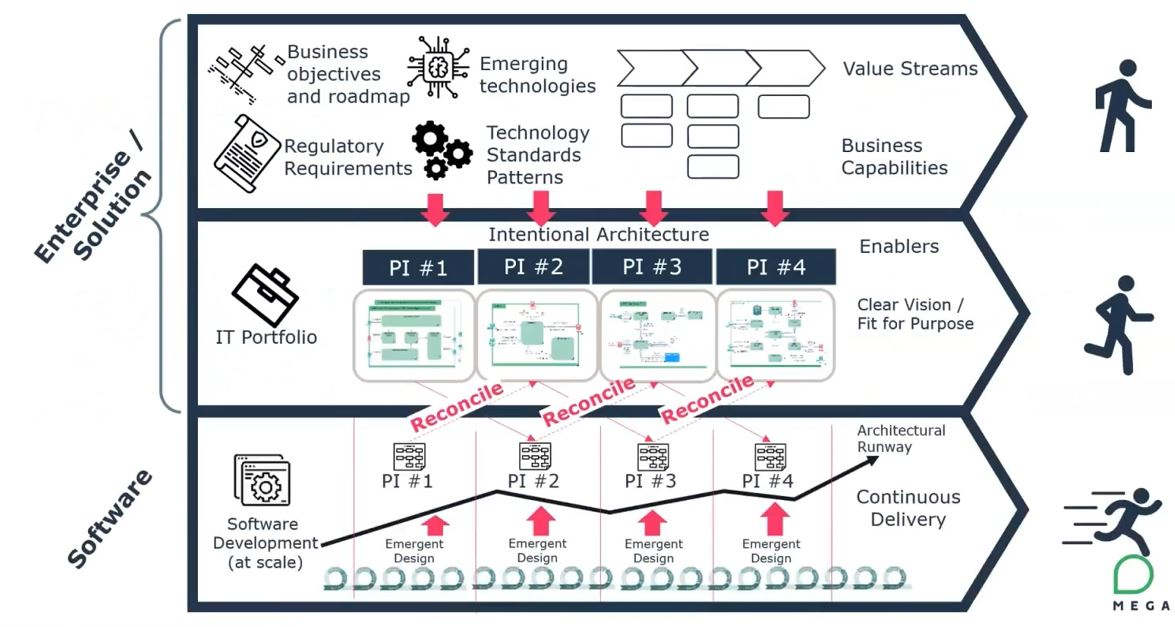


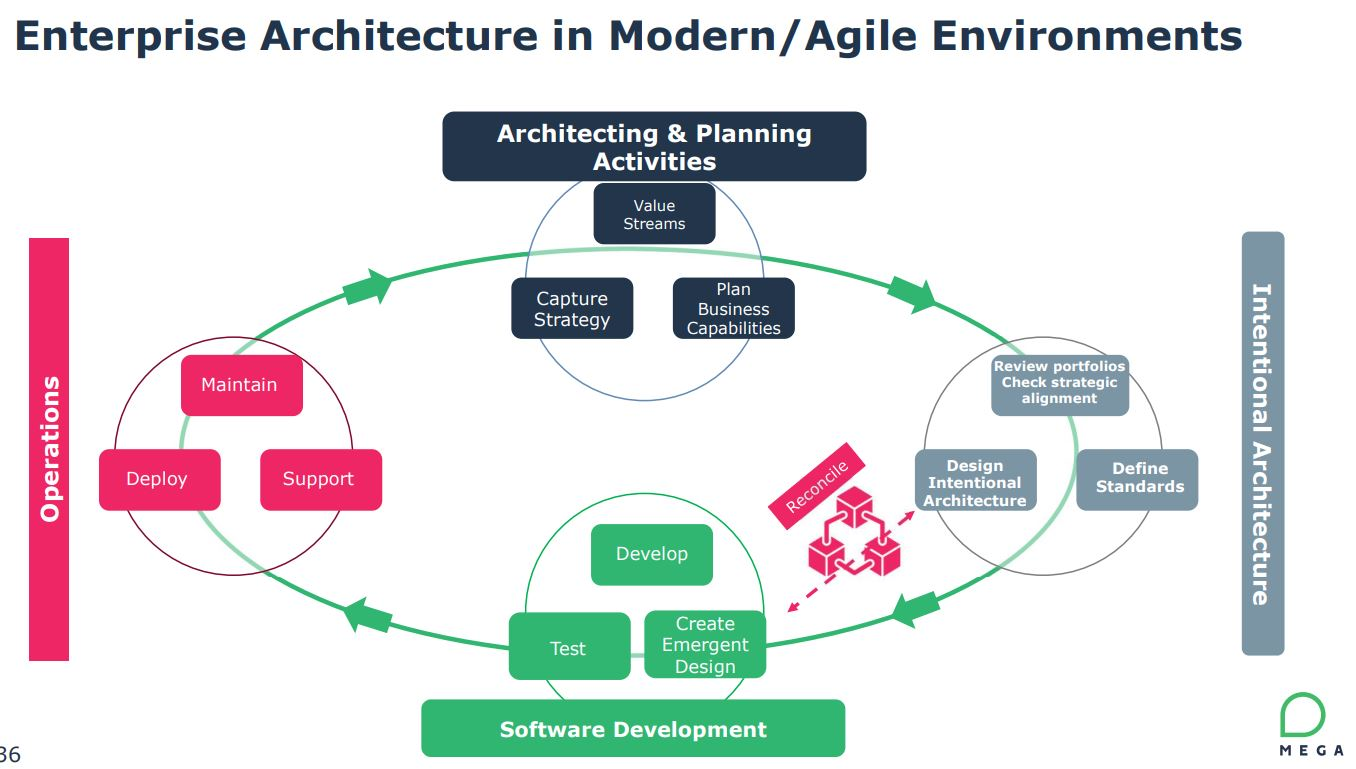
-- Represent major business processes as value streams and align corresponding business capabilities with those processes.

-- Develop corresponding Business Architecture, Application Architecture, Data Architecture, Technology Architecture for the identified value streams & business capabilities when required.

2. Build an Intentional Architecture (Solutions/IT architecture) for the identified ABB's. The IT architecture should be able to fulfill the strategic vision of the organization which has been identified using the Value Streams approach. All IT decisions, technology stack selection, costs should be justified to provide value for all business outcomes

3. During execution of Agile sprints, EA team should collaboratively work with Software architects, designers, dev teams & help them in achieving the Org goals and govern the dev efforts to ensure that the deployment of software artifacts are aligned to the Org strategy and they are able to fulfill the measurable business outcomes.





**IDFC Bank Architecture capability enhancement – Case Study:**

**Enterprise Architecture**

*Value Streams: Online Savings Account Business Process*

Offer Cross Sell VKYC

Perform VKYC

Create Account

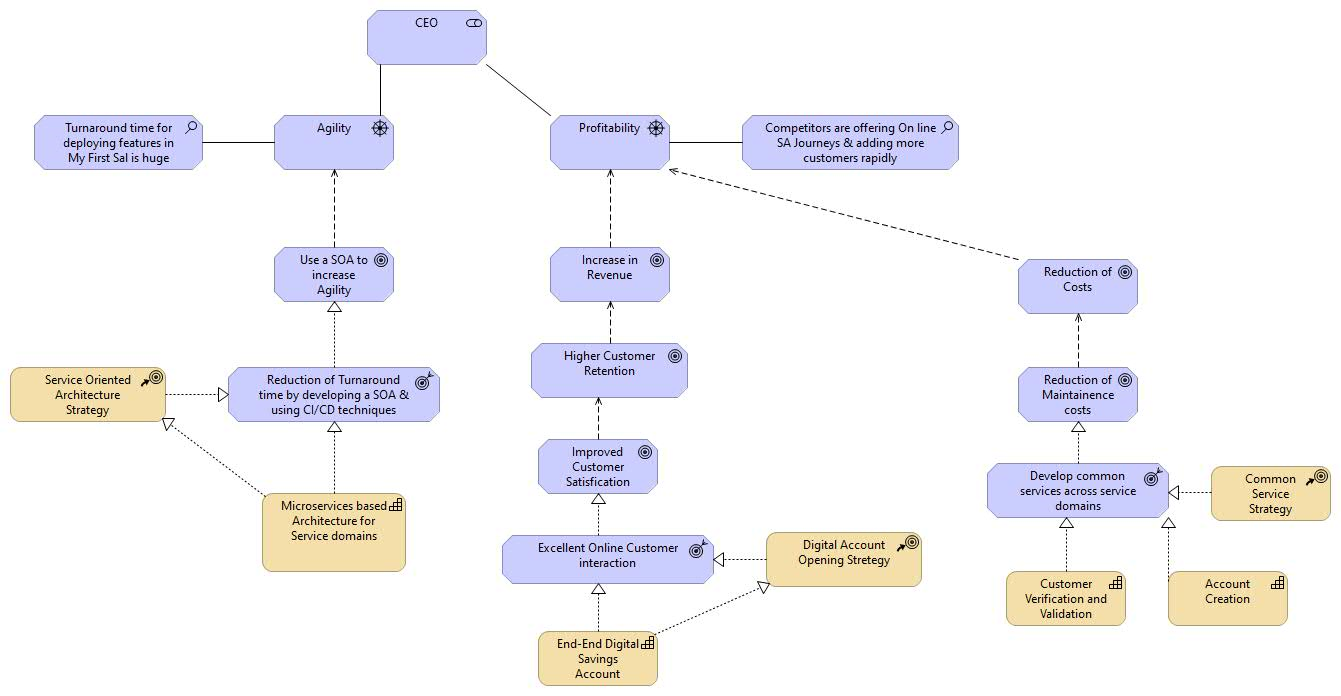
Fraudulent check

Verify Customer & Identification

Account Creation

Customer Verification & Validation

*Business Objectives*:



*Metrics for business outcomes:*

A). Excellent online customer interaction

1. Page loading time of website should be < X seconds

2. Service response time should be within a timeframe of X – Y sec

3. Interactive User Interface with AI powered chat bots for guiding the customer during his Account Opening journey thereby enhancing customer experience. Refer to this requirements document for features needed for implementing the chat bot

4. Error handling to depict User Friendly messages in a way that will guide users to accomplish their task seamlessly despite service downtime. Refer to the complete error handling guide for more details on how different errors need to be handled

5. Ability for the customer to start the journey from where he left off in case he disengages in between. Refer to the following requirements documentation for Journey resumption

6. UI/UX flow to provide a pleasant experience to the customer during his journey. Refer to the documentation that contains VD’s depicting different UI flows for the savings account journey

7. The capability should also support Corporate Accounts (Corporate Salary)

B). Develop common services across service domains

1. The business capabilities “Customer Verification & Validation” & “Account Creation” should be common across the “Savings Account” and “Consumer Loan” service domains. They should be adaptable to changes in the behavior depicted by both these service domains when in action
2. The Services should be able to support the following channels – Web, Tablet, Salesforce CRM, Telephony. All these channels could depict different behavior wrt the process orchestration flow. The solution designed should be flexible to adapt to these changes quickly and be flexible for adaptation when more channels are added in future
3. The services should be able to respond within a time frame of X – Y sec for an initial load of Z concurrent users & scale to accommodate a B% of increase in load on a quarterly basis

*Road Map*:

|  |  |  |
| --- | --- | --- |
| Business Capability | Current State Architecture | Target State Architecture |
| Microservices based architecture for Service domains | Monolith Architecture existing for different Service Domains | 1. Build service capabilities for different service domains based on the identified Value Streams 2. Build a DevOps platform offering CI/CD techniques that will help deliver service capabilities quickly and reduce turnaround time 3. Identify / Architect a container orchestration platform for deploying the microservices |
| End-End Digital Savings Account | My first Salary app existing as a monolith application | Build service capabilities for the “Savings Account” service domain based on the identified Value Streams. These services should also support “Corporate Accounts” |
| Customer Verification & Validation | My first Salary app existing as a monolith application | Build common service capability for “Customer Verification & Validation” identified in the Value Streams. They should support both the “Savings Account” and “Consumer Loan” service domains |

*Emerging Technologies:*

*Technology Standards & patterns:*

*Regulatory Compliance:*

1. PII data encryption In Transit & at rest

**Solutions (Intentional) Architecture**

*Prompt engineering:*

“I work for an Indian Private Bank & I would like to create an Online Savings account as a part of my Digital enablement strategy. Following are my business outcomes for the same

A). Excellent online customer interaction

1. Page loading time of website should be < 5 seconds

2. Service response time should be within a timeframe of 2-5 seconds

3. Interactive User Interface with AI powered chat bots for guiding the customer during his Account Opening journey thereby enhancing customer experience

4. Error handling to depict User Friendly messages in a way that will guide users to accomplish their task seamlessly despite service downtime.

5. Ability for the customer to start the journey from where he left off in case he disengages in between.

6. UI/UX flow to provide a pleasant experience to the customer during his journey

7. The capability should also support Corporate Accounts (Corporate Salary)

B) Reduction of turnaround time

I need to be able to quickly develop and deploy new features essentially with a capability of a production rollout each day increasing Agility.

C). Develop common capabilities across service domains

1. The Business Process (steps) to open a savings account are as follows
   1. Verify Customer & Identification
   2. Perform Fraudulent Check
   3. Create Account
   4. Perform online Video KYC
   5. Offer Cross sell products like Mutual Funds, Term Insurance etc.
2. I need to create capabilities like “Customer Verification & Validation” & “Account Creation” that should work across the “Savings Account” and “Consumer Loan” service domains. They should be adaptable to changes in the behavior depicted by both these service domains when in action
3. Each capability will have a business process orchestration. “Customer Verification & Validation” will have Aadhaar verification, Lead creation, Aadhaar – PAN match, Customer Dedupe, Mobile dedupe. They will connect to third-party services like UIDAI, NSDL and SORs like CBS, MDM, DMS
4. The capabilities should be able to support the following channels – Web, Tablet, Salesforce CRM, Telephony. All these channels could depict different behavior wrt the business process orchestration flow. The solution designed should be flexible to adapt to these changes quickly and be flexible for adaptation when more channels are added in future
5. The capabilities should be able to respond within a time frame of 2-5 seconds for an initial load of 100 concurrent users & scale to accommodate a 20% of increase in load on a quarterly basis

Provide me with a Target State Architecture for the abovementioned business outcomes

*Target state Architecture:*

**1. Presentation Layer:**

* **Description:** The user interface and user experience layer that interacts directly with customers.
* **Components:**
  + Web Application Interface
  + Mobile Application Interface
  + Chatbot Interface
  + Salesforce CRM Integration
  + Telephony Interface

**2. Application Layer:**

* **Description:** The layer responsible for business logic, orchestration, and interaction between different modules.
* **Components:**
  + Microservices for Customer Verification & Validation
  + Microservices for Account Creation
  + Microservices for Fraudulent Check
  + Microservices for Video KYC
  + Microservices for Cross-selling
  + Chatbot Logic

**3. Integration Layer:**

* **Description:** Manages communication with external services, APIs, and third-party systems.
* **Components:**
  + Third-party Integrations (UIDAI, NSDL)
  + Systems of Record (CBS, MDM, DMS)
  + Centralized Integration Hub
  + API Gateway

**4. Data Layer:**

* **Description:** Manages the storage and retrieval of data needed for the application's operation.
* **Components:**
  + Customer Data Store
  + Account Data Store
  + Transaction Data Store
  + Logging and Monitoring Data Store

**5. Security Layer:**

* **Description:** Ensures the security and compliance of the entire system.
* **Components:**
  + Identity and Access Management
  + Encryption Mechanisms
  + Compliance and Auditing Tools

**6. Infrastructure Layer:**

* **Description:** The underlying foundation that supports the deployment and operation of the entire system.
* **Components:**
  + Cloud Infrastructure (AWS, Azure, etc.)
  + Container Orchestration (Kubernetes)
  + Serverless Components
  + Networking Infrastructure

**7. DevOps and Continuous Integration/Continuous Deployment (CI/CD) Layer:**

* **Description:** Manages the development, testing, and deployment processes for ensuring agility.
* **Components:**
  + CI/CD Pipelines
  + Automated Testing Frameworks
  + Version Control Systems (Git)
  + Deployment Automation Tools

**8. Monitoring and Analytics Layer:**

* **Description:** Monitors system performance, user behavior, and provides insights for continuous improvement.
* **Components:**
  + Application Performance Monitoring (APM) Tools
  + Log Aggregation and Analysis Tools
  + User Analytics Tools

**9. Scalability Layer:**

* **Description:** Ensures that the system can scale horizontally and vertically to handle increasing loads.
* **Components:**
  + Auto-scaling Mechanisms
  + Load Balancers
  + Elastic Infrastructure

**10. Business Process Orchestration Layer:**

* **Description:** Orchestrates the sequence of activities required for key business processes.
* **Components:**
  + Workflow Management System
  + Business Process Automation Tools

**11. AI and Machine Learning Layer:**

* **Description:** Integrates AI and machine learning capabilities for enhanced customer interaction.
* **Components:**
  + Natural Language Processing (NLP) Engines
  + Machine Learning Models for Predictive Analytics

**12. Session Management Layer:**

* **Description:** Manages and maintains user sessions to ensure continuity and security.
* **Components:**
  + Session Handling Mechanisms
  + Secure Token Management

*Possible Technology stack:*

**Web Application:**

* **Frontend:**
  + React.js or Vue.js for building responsive and interactive user interfaces.
  + Redux or Vuex for state management.
  + TypeScript for enhanced code quality and scalability.
* **Backend:**
  + Node.js or Spring Boot for building scalable and high-performance server-side applications.
  + Express.js or Java Spring MVC for handling HTTP requests.
  + GraphQL for efficient data querying.

**Microservices Architecture:**

* **Containerization and Orchestration:**
  + Docker for containerization.
  + Kubernetes for container orchestration and management.
* **Microservices Communication:**
  + RESTful APIs for communication between microservices.
  + Kafka or RabbitMQ for event-driven communication.

**AI-powered Chatbots:**

* **Chatbot Framework:**
  + Dialogflow or Microsoft Bot Framework for building conversational interfaces.
  + Natural Language Processing (NLP) libraries like spaCy or NLTK.

**Error Handling:**

* **Monitoring and Logging:**
  + Prometheus and Grafana for monitoring.
  + ELK Stack (Elasticsearch, Logstash, Kibana) for centralized logging.

**Session Management:**

* **Session Handling:**
  + JSON Web Tokens (JWT) for secure and stateless session management.

**Integration Layer:**

* **API Gateway:**
  + Apigee or Kong for API management and gateway.
  + GraphQL for flexible data querying.
* **Third-Party Integrations:**
  + SDKs for UIDAI, NSDL, and other third-party services.
  + REST or SOAP for interfacing with external systems.

**Data Layer:**

* **Databases:**
  + MongoDB or PostgreSQL for flexible and scalable data storage.
  + Redis for caching.
* **Data Security:**
  + Encryption mechanisms for sensitive data (AES, SSL/TLS).

**Security Layer:**

* **Identity and Access Management:**
  + OAuth 2.0 for identity management.
  + Keycloak or Auth0 for authentication and authorization.

**Infrastructure Layer:**

* **Cloud Services:**
  + AWS, Azure, or Google Cloud for scalable cloud infrastructure.
* **Container Orchestration:**
  + Kubernetes for managing containerized applications.
* **Serverless:**
  + AWS Lambda or Azure Functions for serverless computing.

**DevOps and CI/CD:**

* **Version Control:**
  + Git for version control.
* **Continuous Integration/Continuous Deployment:**
  + Jenkins or GitLab CI for CI/CD pipelines.
  + Docker Hub or AWS ECR for container registry.

**Monitoring and Analytics:**

* **Monitoring Tools:**
  + Prometheus for monitoring.
  + Grafana for visualization.
* **Analytics:**
  + Google Analytics or Mixpanel for user analytics.

**Scalability Layer:**

* **Auto-Scaling:**
  + AWS Auto Scaling or Kubernetes Horizontal Pod Autoscaler.
* **Load Balancers:**
  + AWS Elastic Load Balancer or NGINX for load balancing.

**Business Process Orchestration:**

* **Workflow Management System:**
  + Camunda or Apache Airflow for workflow orchestration.

**AI and Machine Learning:**

* **Machine Learning Frameworks:**
  + TensorFlow or PyTorch for machine learning capabilities.

**Session Management Layer:**

* **Session Handling:**
  + Redis for in-memory data store.
  + JWT for token-based session management.

**Mobile Application (Optional):**

* **React Native or Flutter:**
  + For cross-platform mobile application development.

*Target state Architecture (AWS):*

### Target State Architecture for Hybrid Cloud with AWS and On-Premise Core Banking Systems:



#### **Components and Considerations:**

1. **On-Premise Core Banking Systems:**
   * The existing core banking systems remain on-premise, ensuring data residency and compliance with regulatory requirements.
   * Implement secure communication channels, such as VPNs or dedicated leased lines, to connect on-premise systems with the cloud.
2. **Hybrid Cloud Connectivity:**
   * AWS Direct Connect: Establish a dedicated network connection from on-premise data centers to AWS. This provides a private and high-bandwidth connection, improving performance and security.
   * AWS VPN: Utilize AWS VPN services for secure communication between on-premise and AWS environments over the internet.
3. **Amazon VPC (Virtual Private Cloud):**
   * Create an isolated network environment within AWS using Amazon VPC to host cloud resources securely.
   * Implement multiple subnets for organizing resources and controlling access.
4. **Public and Private Subnets:**
   * Design the VPC with public subnets for resources that need to be accessible from the internet (e.g., web servers) and private subnets for sensitive backend systems.
5. **AWS Managed Services:**
   * Leverage AWS managed services such as Amazon RDS for databases, Amazon S3 for storage, and Amazon ECS for container orchestration.
   * Utilize managed services to offload operational overhead and ensure scalability.
6. **Elastic Load Balancing (ELB):**
   * Implement ELB to distribute incoming application traffic across multiple Amazon EC2 instances. This improves fault tolerance and ensures high availability.
7. **AWS Lambda for Serverless Computing:**
   * Integrate AWS Lambda for serverless computing to run code without provisioning or managing servers. This can be beneficial for event-driven architectures.
8. **Data Integration and ETL:**
   * Implement AWS Glue or Amazon Kinesis for data integration and ETL (Extract, Transform, Load) processes. This facilitates seamless data movement between on-premise and cloud environments.
9. **Security and Identity Management:**
   * Utilize AWS Identity and Access Management (IAM) for access control and user management.
   * Implement AWS Key Management Service (KMS) for encryption of data at rest and in transit.
10. **Monitoring and Logging:**
    * Implement AWS CloudWatch for monitoring resources and applications in real-time.
    * Utilize AWS CloudTrail for auditing and tracking user activity within the AWS environment.
11. **Disaster Recovery and Backup:**
    * Implement AWS Backup and design a disaster recovery plan to ensure data resilience and business continuity.
    * Explore AWS services like AWS Backup and Amazon S3 versioning for data backup and recovery.
12. **Compliance and Governance:**
    * Adhere to regulatory compliance standards applicable to the banking industry.
    * Implement AWS Config for monitoring and managing compliance.
13. **DevOps Practices:**
    * Adopt DevOps practices for continuous integration and continuous delivery (CI/CD) using AWS CodePipeline and AWS CodeBuild.
14. **Hybrid Cloud Strategy:**
    * Define a clear hybrid cloud strategy that addresses workload placement, data residency, and application performance requirements.
15. **Customer-Facing Applications:**
    * Host customer-facing applications in the public cloud to take advantage of scalability and global reach.
    * Implement AWS CDN (Content Delivery Network) services for efficient content delivery.
16. **Customer Data Security:**
    * Implement encryption for customer data both at rest and in transit.
    * Leverage AWS WAF (Web Application Firewall) for protecting against web application attacks.
17. **API Gateway:**
    * Implement Amazon API Gateway to manage and secure APIs, facilitating communication between on-premise systems and AWS services.
18. **Integration Hub:**
    * Consider an Integration Hub (Integration Platform) to facilitate seamless communication and data exchange between on-premise core banking systems and AWS services.
19. **Multi-AZ Deployments:**
    * Design multi-Availability Zone (AZ) deployments for critical systems to ensure high availability and fault tolerance.
20. **Training and Governance:**
    * Provide training to the IT teams on AWS services, best practices, and security measures.
    * Establish governance policies to manage resource provisioning and access control.

**Innovation Idea**

**Establishing an EA practice – Group Activities**

* EA practice team members to act as consultants for different projects that requires the role of Enterprise Architecture.
* Team to collate different case studies on how they delivered real business value to their clients based on their project experiences. These collaterals will be helpful in projecting our expertise while responding to RFPs
* Team to brainstorm on preparing a common framework that can be used across various projects while consulting
* Team to brainstorm & prepare various governance checklists that will help them to monitor the implementation and guide Agile teams in delivering capabilities that are aligned with the business outcomes.
* Team to brainstorm on how Gen AI can assist architects in building the Solutions/Application architecture layers, determining appropriate technology stacks that will help deliver the business outcomes
* Team to gather for monthly presentations on emerging trends in Enterprise Architecture, Technologies
* For client consultations (Prior to floating RFP’s - where the client’s vision is not clear), we will need domain experts to study emerging market trends in various domain processes and identify digital transformation initiatives (blueprints) that can help shape the client’s vision. E.g. if the client’s main business is to provide loans, the domain consultants will come up with innovative digital transformation solutions that will have a high impact on the client’s business

https://youtu.be/JuEE0MySHY0