FAO
Sola Suite & Open Tenure Assessment

Agenda

Food and Agriculture Organization of the United Nations

1. Introduction
2. Assessment Guidelines
3. Assessment Parameters
4. Assessment Output
5. Technical Recommendation
6. Quality Recommendation
7. What Next

Introduction

Need Of The Assessment
Scope of the Work

Need of the Assessment

Food and Agriculture Organization of the United Nations

- The Land Tenure Unit in the Partnerships and South-South Cooperation Division (DPS) leads FAO's work in support of the Voluntary Guidelines on the Responsible Governance of Tenure of Land, Fisheries and Forests in the Context of National Food Security (VGGT).

- The suite of software applications is based on open source, thereby enabling any user to customize the software to the specific requirements of the country legal, institutional and organizational frameworks

- The objective of the assignment is:
  - To complete an independent quality review of the source code, systems architecture and system documentation of the software applications, recommend improvements, as necessary, and to prepare a plan for Quality certification of the application software with time and cost estimation.

Scope of Work

Food and Agriculture Organization of the United Nations

- External Assessment has been carried out from Zensar for FAO of SOLA Suite of applications to understand the quality of the code and architecture of the existing sola suite application.

- The scope of work is limited to Technical assignment only and does not covered any functional aspect of any project. Following modules involved in the scope
  - Sola Registry : Java Desktop application built upon LADM
  - Sola Systematic Registration : Java Desktop application built upon LADM
  - Sola State Land : Java Desktop application built upon LADM
  - Sola Web Admin : Java Web application built upon LADM
  - Sola Community Server : Java Web application built upon LADM with extension inspired from STDM
  - Sola Open Tenure : Java Mobile application

- Deliverables
  - Installation and Deployment Guidelines [Refer : Annexure 1]
  - Code Review Document [Refer : Annexure 1]
  - Sola and Open Tenure Architecture Document [Refer : Annexure 1]

Assessment Guidelines

Technology Stack Mapping
What has gone wrong ?

Technology Stack Mapping

Food and Agriculture Organization of the United Nations

Component / System	Software / Tools
Sola Registry	• Application Server – Glass Fish, • DB server –PostgreSQL, • GIS server – Geo Server, • Reports – Jasper Reports • Java 7 Security Model
Sola Systematic Registrations	
Sola State Land	
Sola Community Server	
Sola Web Admin	
Sola Open Tenure	• Application Server – Apache HTTP client, • DB server –H2 DB, • GIS server – Google Maps for Android • Reports – Java Reports & • Android Platform Security Model

What has gone wrong?

Food and Agriculture Organization of the United Nations

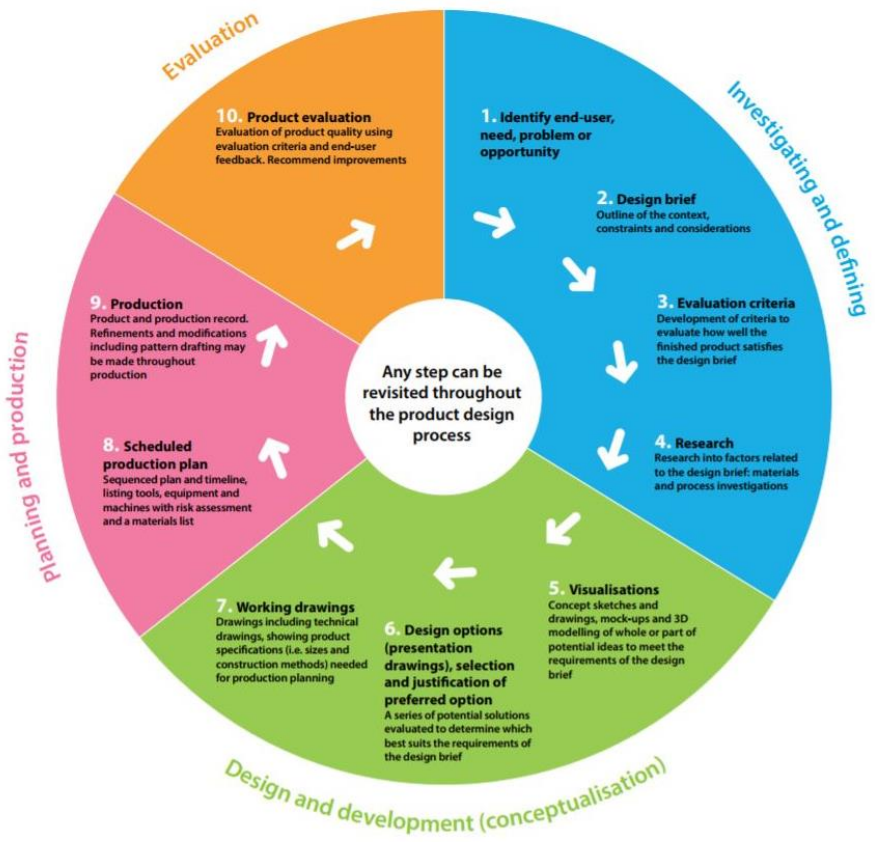
Challenges / Issues	Priority	Severity	Solutions Provided	Remarks
Code Quality of the applications in SOLA Suite	Medium	High	Need to implement Java Standard coding and other code quality standards practices in all the SOLA application Suite.	Re- architecture is required, but not finalized.
Unit Test coverage	Medium	Medium	Unit test coverage is calculated on the basis of Junit test cases written in the	Plans provided to FAO
Overall Quality improvement in code from technical and architectural view point.	Critical	High	To improve quality standards, the result of tools and manual code review observations will be taken into consideration. All the issues and warnings will be resolved.	Zensar has provided the estimation and plan for this solution.
Java Standard coding practices	High	Critical	Need to implement Java Standard coding practices in all the SOLA application Suite.	Zensar has provided the estimation and plan for this solution.
Architectural Modernization of applications	High	Medium	Application restructuring can be done with modern architecture and better maintainability for all the applications.	Zensar has provided the estimation and plan for this solution.

Assessment Parameters

Process for Evaluation

Evaluation Process

Food and Agriculture Organization of the United Nations



Following points are covered in assignment :

- All the code for 6 applications taken from GITHUB for code review purpose.
- Manual code review is done to ascertain the quality standards followed in the system.
- External Automated Code review tools used are as follows :
  - PMD Checkstyle
  - FireBug
  - J-Architect
  - SonarQube
- Architectural documents were taken into consideration for Architectural review.
- Suggestions and recommendations added
- After code analysis and architectural documents, Overall external summary has been prepared

Assessment Output

Assessment Output

Food and Agriculture Organization of the United Nations

Sr.No	Detailed Information	Sola Registry	Sola Systematic Registration	Sola State Land	Sola Community Server	Sola Web Admin	Open Tenure
1	Code Complexity	HIGH	MEDIUM	MEDIUM	LOW	MEDIUM	HIGH
2	Hard Coding	HIGH	HIGH	HIGH	HIGH	HIGH	HIGH
3	Error Handling	MEDIUM	MEDIUM	MEDIUM	MEDIUM	MEDIUM	MEDIUM
4	Security	MEDIUM	MEDIUM	MEDIUM	MEDIUM	MEDIUM	MEDIUM
5	Code Comments	LOW	LOW	LOW	LOW	LOW	LOW
6	Unit Test Coverage	LOW	LOW	LOW	LOW	LOW	LOW
7	Issues - Sonar	2000+	2000+	2000+	1699	1019	2000+
8	Duplicate Code	HIGH	MEDIUM	HIGH	LOW	LOW	HIGH
9	Critical Issues	25	25	26	15	14	50+
10	Overall Code Quality	LOW	MEDIUM	MEDIUM	MEDIUM	MEDIUM	LOW
11	Architecture Changes	REQUIRED	REQUIRED	REQUIRED	REQUIRED	REQUIRED	REQUIRED
12	Technical Debts	HIGH	MEDIUM	HIGH	HIGH	MEDIUM	MEDIUM

Technical Recommendation

Coding Principles
Architecture Principles
Recommendations Approach
Architecture Pattern

Coding Principles

Food and Agriculture Organization of the United Nations

Following are the Coding Principles are/will be applied for to the application Sola Suite System

- **Open Closed Design Principle** – open for extension close for modification
- **Single Responsibility Principle (SRP)** – one class should do once thing and do it well
- **Dependency Injection or Inversion principle** – Don't ask for objects, Let framework gives you
- **Favor Composition over Inheritance** – Code reuse without cost of inflexibility
- **Liskov Substitution Principle (LSP)** – Sub type must be substitutable for super type
- **Interface Segregation Principle (ISP)** – Avoid monolithic interface which reduce pain on client side
- **Programming for Interface not implementation** – Helps in maintenance and improves flexibility
- **DRY (Don't Repeat Yourself)** – avoid duplication of the code
- **Encapsulate What changes** – hides the implementation details , Helps in the maintenance

Architecture Principles

Food and Agriculture Organization of the United Nations

Following are the Design Principles are/will be applied for to the application Sola Suite Application System :

- Separation of concerns
- Modularity based on functional abstractions and layers
- Single Responsibility Principle
- Don't repeat yourself (DRY)
- No Point to Point integration
- Interoperability by leveraging standard **:** The proposed architecture must support recognized standards and avoid proprietary implementations, interfaces, protocols, APIs or add-ons.
- Data Duplication/Replication
- Scalability: The architecture will be designed and built to cater to demands of Scalability. Scalability is the capability to expand the capacity of the solution gracefully, i.e. without having to disrupt the existing solution, or without incurring prohibitive costs.
- Use cloud based offering by default and not exception– Explore & Use AWS based cloud services for new system. Exception deployed outside of cloud will be done with justification and approval.
- Security: The architecture must have built-in features to ensure that proper levels of security are applied and catered for according to set security standards and guidelines.

Recommendation Approach : 1

Food and Agriculture Organization of the United Nations

Technical Quality Improvement : Improve all the warnings and bugs appeared in the various tools and manual review .

What we are going to achieve ?

- Coding standards compliance
- Maintainability and extensibility—fixing, updating and improving software should be as simple as possible, not inherently complex
- Readability, consistency—how easy it is to read and understand sections of the code; this includes code clarity, simplicity, and documentation.
- Predictability, reliability, and robustness—software behavior should be predictable, and not prone to hidden bugs.
- GENERAL PARAMETERS
  - Use checked exceptions for recoverable conditions
  - Favor the use of standard exceptions
  - Return empty arrays or collections, not nulls
  - Inappropriate visibility / accessibility of classes and members
  - Adhere to generally accepted naming conventions
- Comment – satisfy 70% comment criteria
- Complexity - branching statements - 1-4 low , 5-7 for moderate , 8-10 for high and 11+ very high
- Cohesiveness of Class
- Proper resource utilization and release , Use of configuration files, constants file to remove hard coding
- Logger configuration and proper usage
- Test case coverage and Average LOC per class

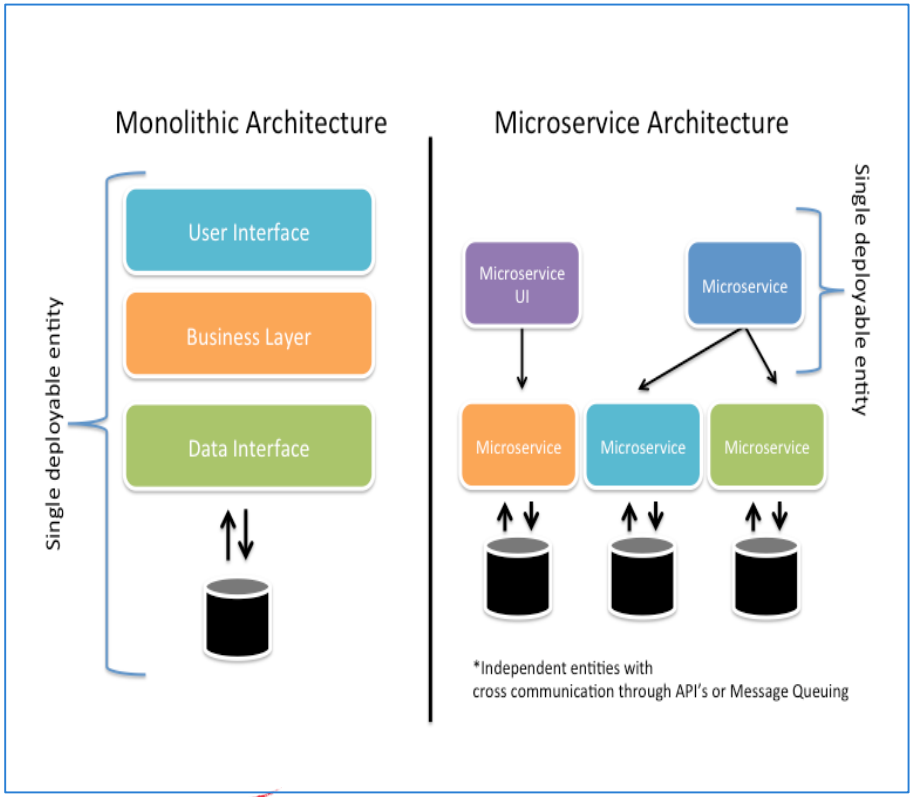
Modules	Efforts (PD) *
Sola Registry	115
Sola Systematic Registration	130
Sola State Land	135
Sola Web Admin	70
Sola Community Server	80
Sola Open Tenure	92
Total Efforts (Development)	**622**

* Only Development Efforts are considered.

Recommendation Approach : 2

Food and Agriculture Organization of the United Nations

Long Term Strategy : Revamp of old project and modernize the code as per new architecture



Modules	Efforts (PD) *
Sola Registry	235
Sola Systematic Registration	250
Sola State Land	265
Sola Web Admin	145
Sola Community Server	155
Sola Open Tenure	205
Total Efforts (Development)	**1100**

* Only Development and Unit Testing Efforts are considered. Refer below for project timeline

Several microservices frameworks that we can use for developing for Java. Some of these are:

- Spring Boot: This is probably the best Java microservices framework that works on top of languages for Inversion of Control, Aspect Oriented Programming, and others.
- Jersey: This open source framework supports JAX-RS APIs in Java is very easy to use.
- Swagger: Helps you in documenting API as well as gives you a development portal, which allows users to test your APIs.

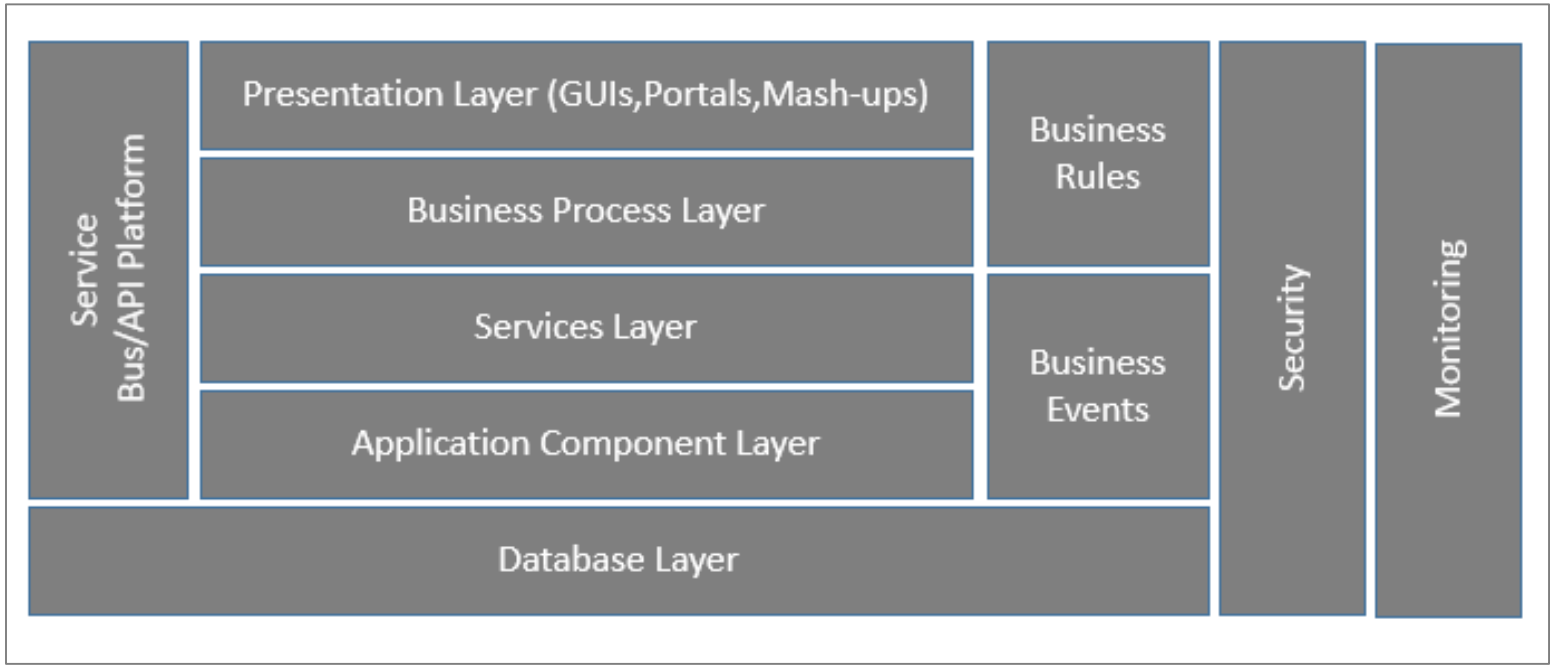
Consider include: Dropwizard, Ninja Web Framework, Play Framework,

RestExpress, Restlet, Restx, and Spark Framework.

Architecture Pattern : Approach 2

Food and Agriculture Organization of the United Nations

Patterns will support reuse of software architecture & design and will also help improve software quality and reduce development time



Quality Recommendation

Framework – Purposeful Open Source
Software Quality Certification
Component Responsibility Matrix
Key Solution Tenets
Open Source Score Card
Open Source Check List

Framework – Purposeful Open Source

Food and Agriculture Organization of the United Nations

	B2B(Recommend ation)	Multi Vendor	Rocket Ship to Mars	Controlled Eco-System	Mass Market	Speciality Library
Main Benefits	Driving industry adoption of your technology	Collaboration with partners; address a set of shared problems	Quick, focused effect in a specific area	Can build a sustainable ecosystem in which founding organization has strong influence	Large user base can make project broadly influential	Ensure quality solution to a specific problem;
Main Drawback	Little or no collaborative Development	Sometimes off-putting to individual Contributors	Collaboration only available from those who share a very specific vision	Compromise needed to avoid forks (Commercial)	Huge user base needs filtering for dev Community	High barriers to entry; relatively small developer Pool
Development Speed	Fast; pace set by business goals	Usually moderate, but depends on needs of Participants	Fast; escape Velocity	Medium	Slow medium; swift change destabilizes user Base	Gets slower over time, as library stabilizes

Framework – Purposeful Open Source

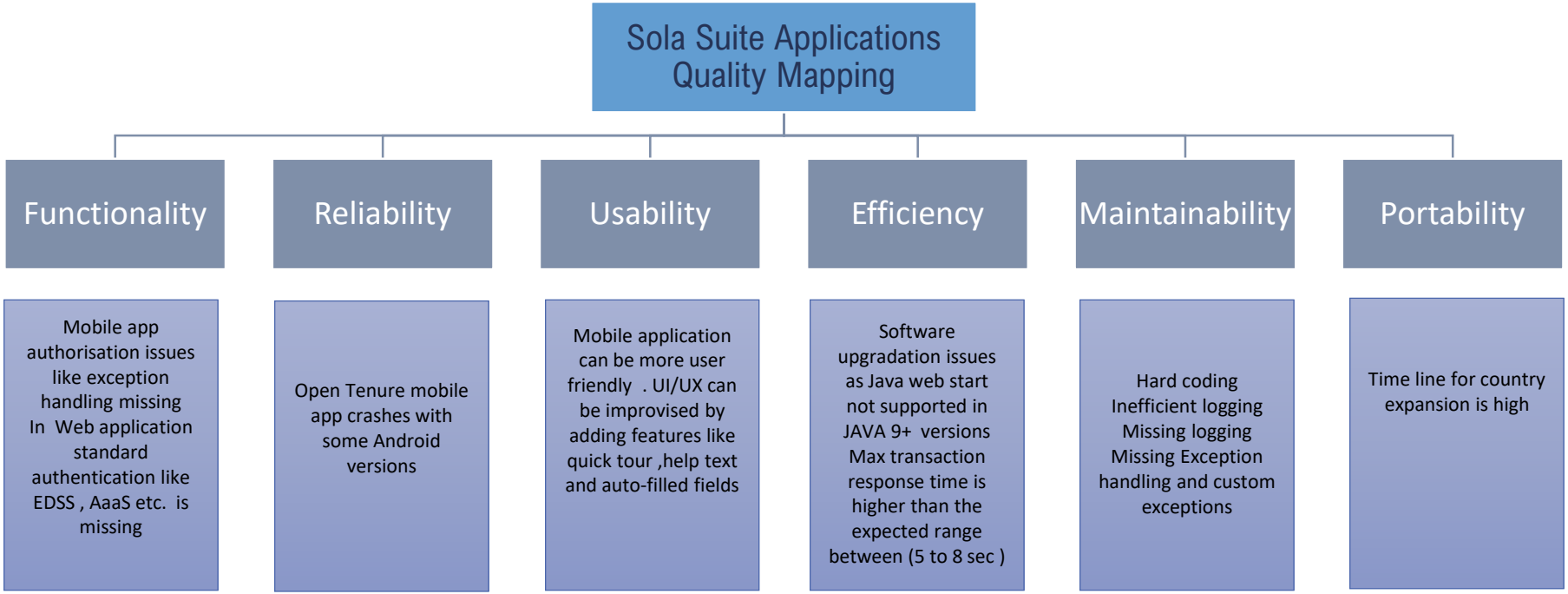
Food and Agriculture Organization of the United Nations

	B2B	Multi Vendor	Rocket Ship to Mars	Controlled Eco-System	Mass Market	Speciality Library
Community Standards	Oriented toward Organizations	Welcoming but formal; difficult for individuals	Focused on core Group	Welcoming, with some onboarding Structures	Fully open; scales via users helping users	High barrier; contributors need to be Experts
Typical Licensing	Almost always non-copyleft	Usually non-copyleft, because many internal forks	Usually non-copyleft, but with occasional exceptions	Any, but requires thought re-plugins	Usually non-copyleft, but depends on business strategy	Usually no-copyleft
Open Source Success	Adoption by target partners Successful projects built around core project	Longevity of contribution by committers Variety of organizations Contributing	Speed of development Adoption by target users Achieving original technical Goals	Adoption by target users Increase in number of extension developers	Awareness among users that project is open source Effective filtering of user feedback to developers	High quality of contributors High quality of code

Software Quality Certification

Food and Agriculture Organization of the United Nations

Gaps identified in application with respect to the ISO 25000 standards



Component Responsibility Matrix: Approach 2

Food and Agriculture Organization of the United Nations

Micro Services	API Management	Experience Assembler	Service Gateway
Versioning	Used as Channel Strategy rather than Integration Strategy	Assembles the experience by combining presentation & representation	Act as Proxy for Microservice's Authentication and Authorization
Single Responsibility Principle	Used Primarily as Integration touch point for Mobile Interfaces	Isomorphic design, Support headless browsing	Expose Coarse grained end points exposed via HTTP
Technology Agnostic (Exposed thro' Interfaces)	No Orchestration built / should be built	Delegates all service layer calls to Service Gateway(wherever Orchestration is needed)	Doesn't hold state and will be responsible for Service Composition & Orchestration
No Intra-Service Communication encouraged	No Transformation logic built / should be built	Should not be used to call Microservice layer directly No Business Logic	Should not be exposed to Public Access via interface

Key Solution Tenets

Food and Agriculture Organization of the United Nations

Alignment with FAO Enterprise Digital Architecture	Future-proof Design to expand for newer Channels	Feature-Rich Mobile Solution as unique FAO Brand
Security	Reusability	Scalability
Customer Experience Journey Driven Design	Customized Execution model for faster Delivery with continuous reviews	Continuous Integration, Continuous Deployment and Test Automation

Open Source Score Card

Food and Agriculture Organization of the United Nations

	Sola Registry	Sola Systematic Registration	Sola State Land	Sola Web Admin	Sola Community Server	Sola Open Tenure
License						
Documentation	✓	✓	✓	✓	✓	✓
Code of Conduct	✓	✓	✓	✓	✓	✓
Contributing Guidelines	✓	✓	✓	✓	✓	✓
Open pull requests	✓	✓	✓	✓	✓	
Naming and branding						

Open Source : Pre-launch checklist

Food and Agriculture Organization of the United Nations

Documentation	
Project has a LICENSE file with an open source license	
Project has basic documentation (README, CONTRIBUTING, CODE_OF_CONDUCT)	
The name is easy to remember, gives some idea of what the project does, and does not conflict with an existing project or infringe on trademarks	
The issue queue is up-to-date, with issues clearly organized and labeled	

Code	
Project uses consistent code conventions and clear function/method/variable names	
The code is clearly commented, documenting intentions and edge cases	
There are no sensitive materials in the revision history, issues, or pull requests (for example, passwords or other non-public information)	

Organization	
Marketing plan for announcing and promoting the project	
Someone is committed to managing community interactions (responding to issues, reviewing and merging pull requests)	
At least two people have administrative access to the project	

Way Next

Zensar Recommendation
Proposed Architecture
Proposed Technology Stack
Transition Map
Roadmap / Timelines
Executive Methodology
High Level Testing Solutions
Governance Model
Team Structure
Zensar Digital Full Stack Service Offerings
Commercials

Zensar Recommendation : Approach 2

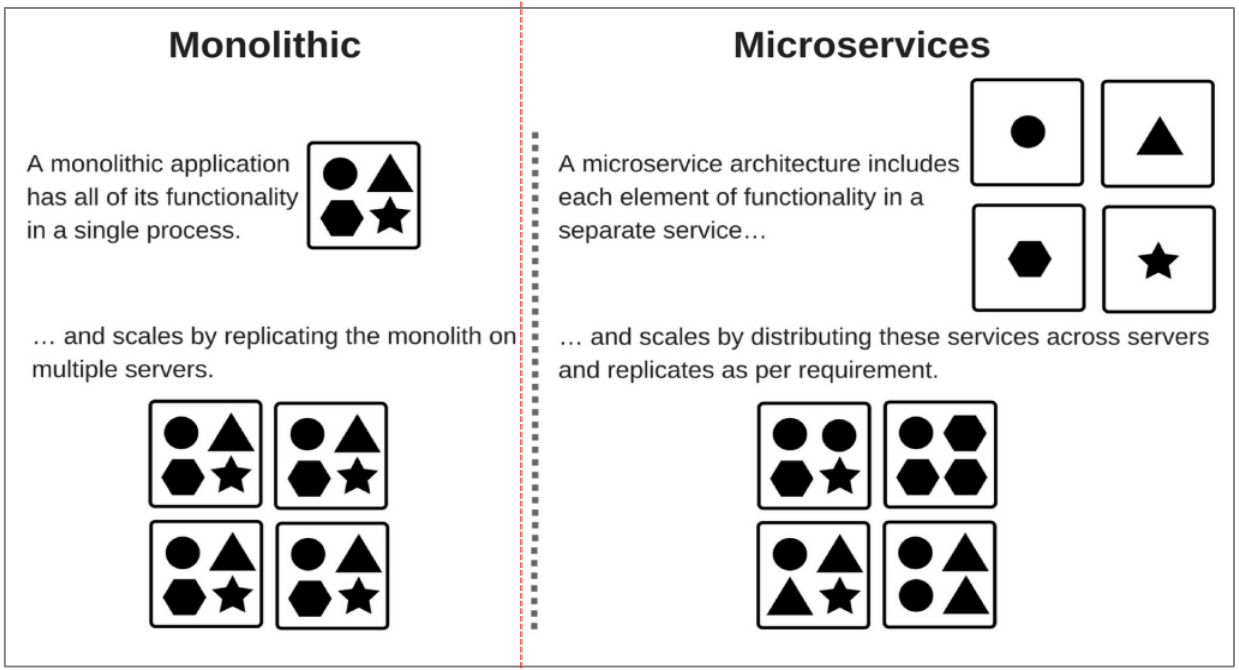
Food and Agriculture Organization of the United Nations

Application architecture will be construct across following layers (one or other layer might be truncated based on final requirement scope & layer applicability) nonetheless the representation is comprehensive one the way application architecture will be construct for future extendibility shake:

- Presentation layer
- Business processes
- Services
- Application Components
- Database

Orthogonal services will be:

- Service bus/API Platform
- Business rules engine
- Business eventing services
- Security
- Business process monitoring



Why: Modernize the existing software ?

Food and Agriculture Organization of the United Nations

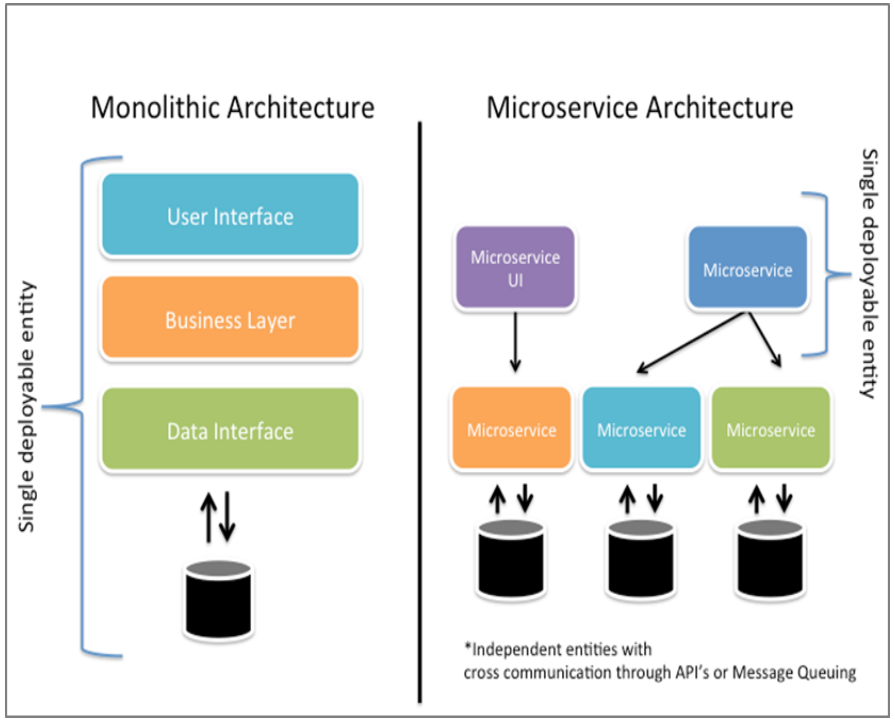
We can modernized the existing application suite with help if new modern architecture – Spring Boot / Microservices which will help as follows.

- **Starts faster than a monolithic** – scope is smaller than a monolithic. This leads to a smaller number of written classes and third-party libraries which must be archived. As a result, the deployment and the start-up are faster.
- **Scale Independently** – A microservice can scale independently using X-axis cloning and Z-axis partitioning based upon their need. This is different from monolithic applications that may have distinct requirements and yet must be deployed together.
- **No long-term commitment to any stack** – you can give greater flexibility to the definition of the language and stack that is best suited for a microservice. Even if you want to restrict the choice of technology, you're not penalized because of past decisions. It enables rewriting the service using better languages and technologies.
- **Independent and frequent Deployments** – each microservice can be independently deployed and redeployed again, without impacting the overall system.

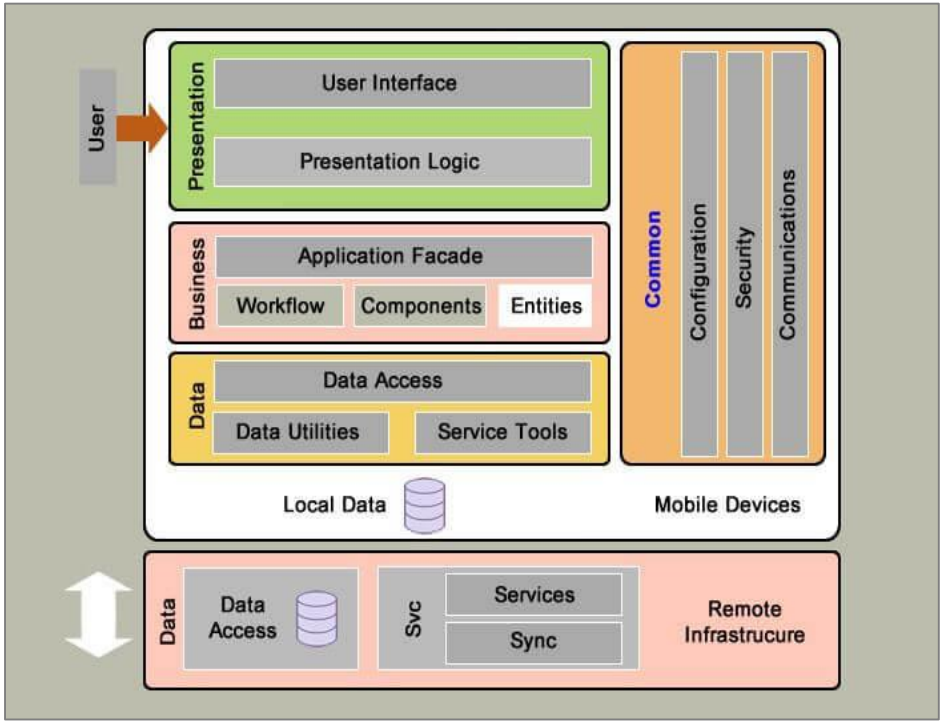
Proposed Architecture

Food and Agriculture Organization of the United Nations

Sola Application Architecture



Open Tenure Application Architecture



Proposed Technology Stack

Food and Agriculture Organization of the United Nations

Integrated Development Environment	Eclipse
Platform	Java 8.0 +
Front End	HTML5, CSS3
Web UI Framework	Bootstrap and AngularJS
Application Framework	Spring Framework
Mobile Application	Native
Data Access Layer	Hibernate
Web & Application Server	Apache HTTP server & tomcat
Rule Engine	Drool (when required)
Operating System	Linux (RedHat)
Build/continuous integration	Jenkins , Gradle / Maven, JUnit
Java source code repository	Bit Bucket
Code Analysis	SonarQube
Performance / Load Testing	Apache JMeter
Task / Issue Tracking	Jira
Collaboration Technologies	Confluence
Hosting Platform	Amazon Web Services (AWS) – If Required

Transition Map

Food and Agriculture Organization of the United Nations

Timeline / Workstream	Due Diligence	Transition Initiation & Planning	Transition Execution	Operational Readiness Gate	Steady State
Set-up & Facilities		✓		✓	
Knowledge Transition	✓	✓	✓	✓	
Transition Governance		✓	✓	✓	
Change Management			✓	✓	✓
Planning for Transformation		✓	✓	✓	✓
Transition Risks & Mitigation	✓	✓	✓	✓	
Transition Mgmt. Tools		✓	✓	✓	

Roadmap / Timelines – Approach 2

Food and Agriculture Organization of the United Nations

Design

All the in-scope features shall be designed and agreed with the stakeholders during this phase.

01

W1

Development

Development of the designed features incorporating industry best practices.

W2 – W15

02

Testing

A regressive testing will be performed before the changes are handed over to UAT team.

03

W16-W18

UAT

Support will be provided during the UAT and this phase will be executed by both FAO and Zensar teams.

W19-W20

04

UAT Fixes

Defect fixes if there are any occurred during the testing phase.

05

W21

Go Live

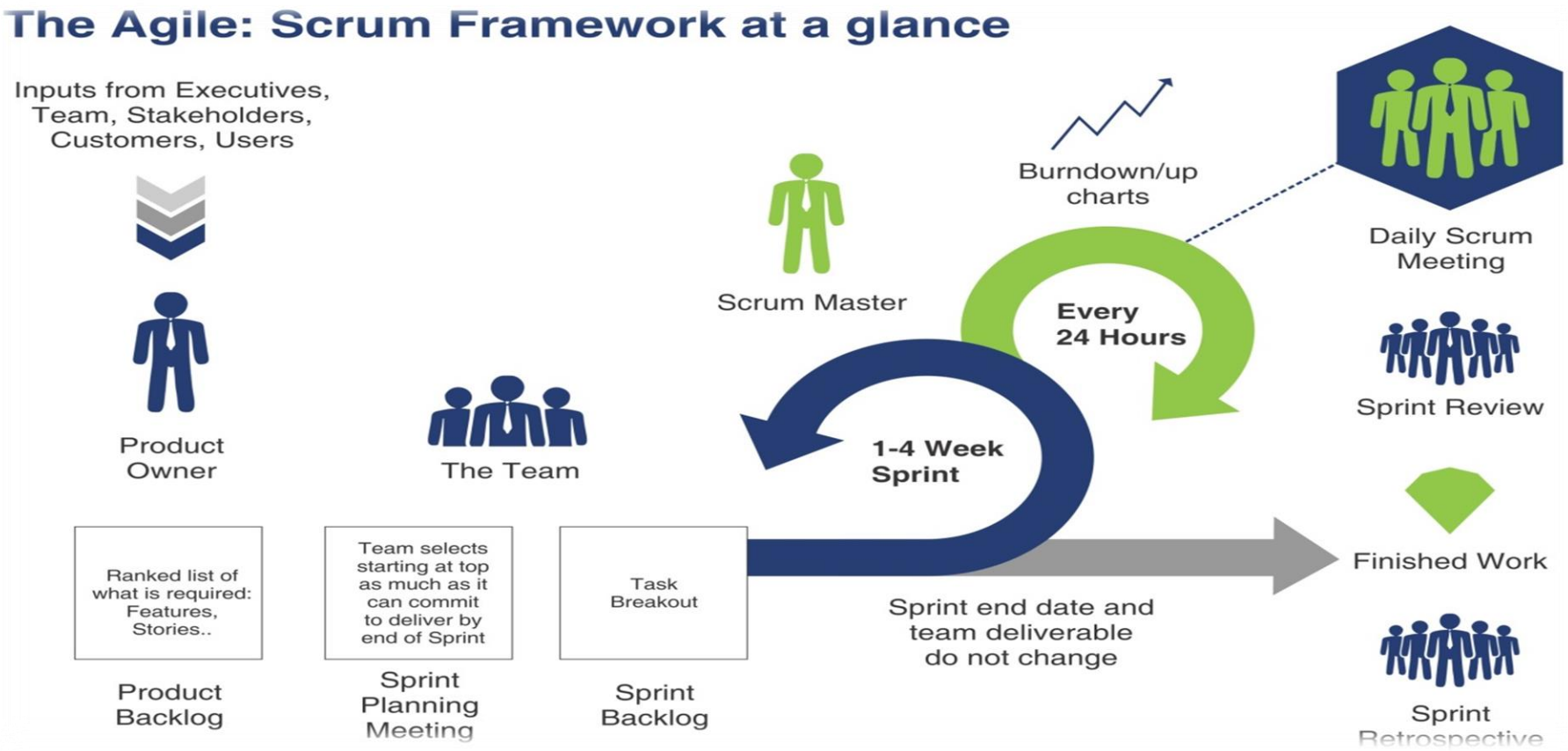
The product shall be available for the production servers and is available for the users to access and use.

W22

06

Execution Methodology

Food and Agriculture Organization of the United Nations



High Level Testing Solution

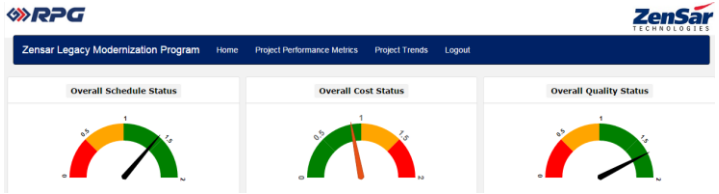
Food and Agriculture Organization of the United Nations

Ensuring 100% Coverage

Web Portal
Consumer | Corporate

Mobile App
Consumer | Corporate

Test Metrics Collection – ensuring Quality Gates



API – Testing including Request/Response using SOAP UI / JMeter

- Digital Channel Mix - Functional Web / Mobile Responsive Testing
- Phone/ Laptops
- Functional Automation
- Portal Performance Testing
- Portal Security Testing – SAST / DAST / RAST
- Cross Browser Testing
- Device Mix - Native Apps / Android Devices / Platforms
- Mobile App – Interrupt Testing
- Mobile App Performance Testing
- Mobile Security Testing

Apache JMeter™ sonarqube Burp suite

Internal Systems (API Testing)

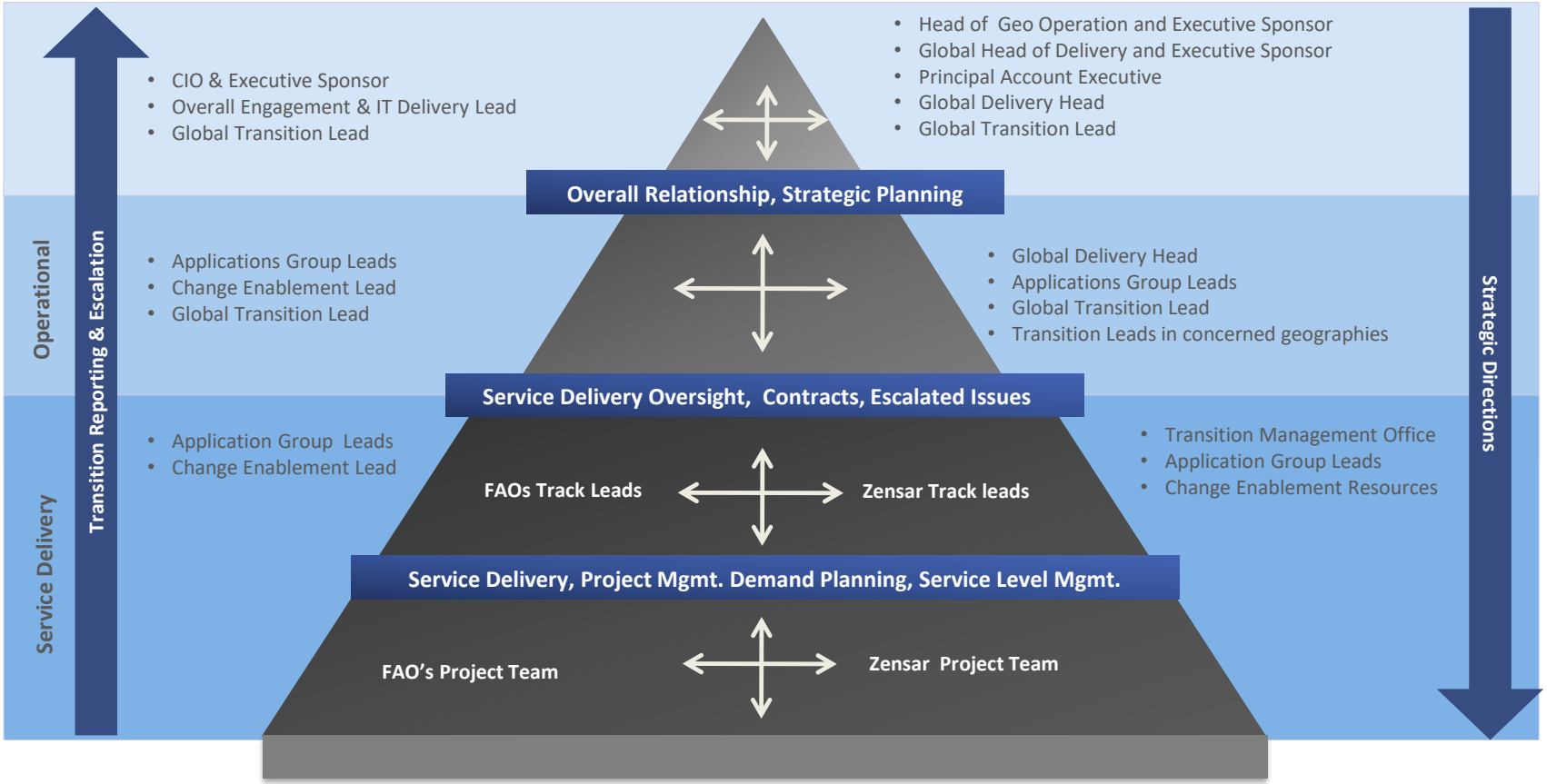
Web API | Rest API | Payment API

Middle Layer Testing RESTful Services

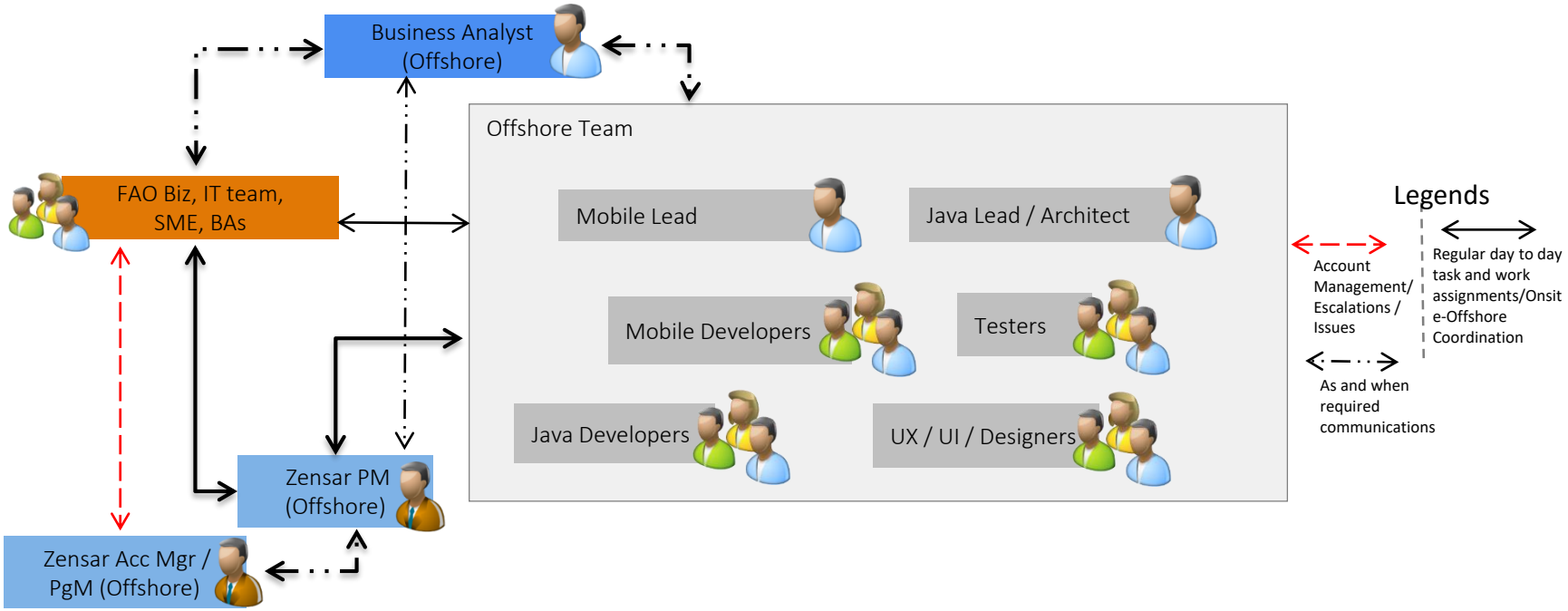
Core Banking System

Governance Model

Food and Agriculture Organization of the United Nations



Team Structure



- Working Model (Onsite Offshore Model)
- Project Manager, Business Analyst, UX/UI Specialist, Java Architect, Mobile Lead, Testers will visit for short duration during different phases of project.
- Primary role of Java/Mobile lead : Work Allocation, Code Reviews, Mentoring Resources and Problem solving, carry out assigned Development support work.

Commercials - WIP

Food and Agriculture Organization of the United Nations

Technical Quality Improvement : Improve all the warnings and bugs appeared in the various tools and manual review

Components	Development Price
Sola Suite Application	$ 109,710
Open Tenure (Android Mobile Application)	$ 19,044

Long Term Strategy : Revamp of old project and modernize the code as per new architecture

Components	Development Price
Sola Suite Application & Open Tenure (Android Mobile Application)	$ 227,700

Thank You